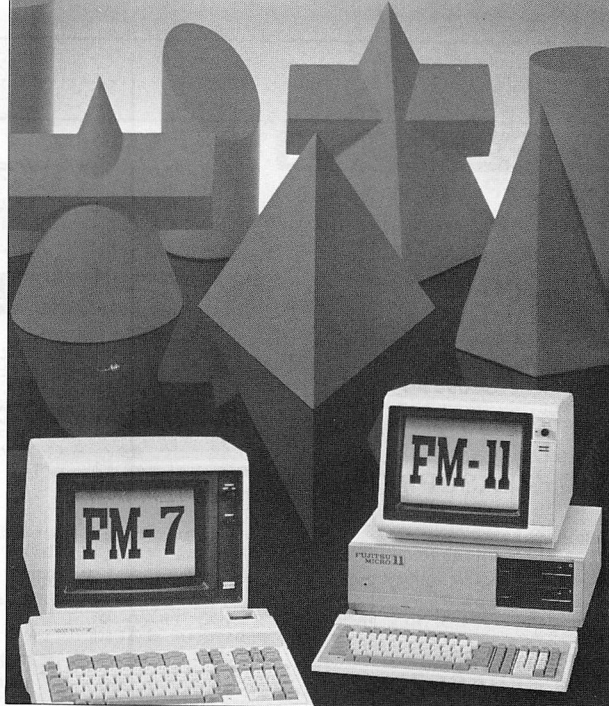
I/O別冊

FM-7/11活用研究

FM-7/11活用研究

CONTENTS

システム/アプリケーション編

FM-11 マイクロコンピュータ支援教育用システム・ソフト…六田嘉明 4
FM-11 画面編集・製図プログラム……六田嘉明 11
フルカラーグラフィック画面ハードコピー… 六田嘉明, 細川治雄 18
予測計算プログラム……松波栄治 27
FM-7 ALL RAM DISK BASIC……コンピ 31
FM-11 グラフィック・エディタ……Yes 33
FILE COPY プログラム……中安博司 63
TortoiseをBASICで……H. KOGCHI 65
TortoiseをKで……小松 仁 72
ディスク・コミュニケーション……ポパイ&ジュリー 79
Kコンパイラ V.3.6 …… COMPAC T & St & Sgn 81
IODOS 9 ……Wonder Soft Sng 93
Kコンパイラ有象無象……COMPAC S 105
Kコンパイラで文字変数を使う……石井正秀 108
FM-7 URA DOS……DOSのTom 113
FM-7 エディタ・アセンブラ↔BASICコミュニケーション… 吉川尚之 121
FM-7 F-BASIC DEBUGGING TOOL……日原洋文 127
FM-7 micro vi……津田伸秀 129
FM-7 Prolog/F V.2……吉川永一 140
FM-7 MUSIC MAKER……COMPAC T 158

ゲーム編

FM-7 Newゴルゴ13……ワンダーソフトSgn 162
FM-7 FLYING SCISSORS……高須晶英 167
FM-7 INDY-7……高畑英樹 171
MELON PANIC(少しの改造で11対応)……三浦 貢 175
FM-7 詰め将棋……土肥一夫 179
FM-7 3D UFO……高畑英樹 190
将棋対局……MAX 193
FM-7 OCTPUS GARDEN……Open-Reach 200
FM-7 バトミントン・ゲーム……吉田 昌 205

資料編

FM-11 サブシステム・コマンド一覧……210
FM-11 ハード仕様……219
FM-11 回路図……246
FM-11 実用ソフト……307

FM-7/8 活用研究正誤表……334

RANDOM BOX

FM-11 KコンパイラのゲームをFM-11で…エンジン・ルームK 17
将棋対局をより面白く……西野直樹 26
『TortoiseをKで』の演算精度をあげる…横井幸彦 64
FM-7 FILES命令の改良……カメ 80
FM-7 "6809デバッガ"のFM-7への移植…COMPAC S 120
FM-7 BASICテキスト文字列サーチ・プログラム…COMPAC S 139
システム開発用簡単エディタEDIT _S……T.T.S 160
FM-7 アプリケーション・ソフト……大森 智 170
Kコンパイラ用三角関数サブルーチン・渡辺誠太郎 178
2^n-1を求める……68T09 S 189
FM-11 FM-8用プログラムのFM-11への移植…浅見 薫 208

注）機種名のないものは7/11対応

I/Oの本

東京・代々木
工学社

ペーパーウェア・シリーズ

ペーパーウェアFP-1100〔1〕
3Dゴルフ・シミュレーション，ザ・麻雀など面白ゲームの他，エディタ・アセンブラ，内部ルーチンの解析記事，全回路図集などを掲載．
B5判140頁　定価1,300円(〒250)

ペーパーウェアPC-9801〔1〕
PC-9801の特徴を活かした日本語データ・ベース，在庫管理，金種計算，給与システムなどオール・ビジネス・プログラム集．PC-8801との共用プログラムも併載．
B5判140頁　定価1,300円(〒250)

ペーパーウェアPC-6001〔1〕
〝BASICコンパイラプログラム・コンテスト〟入選作を一堂に集めたエキサイテング・ゲーム特集号．ソース・リストも全公開．
B5判148頁　定価1,300円(〒250)

ペーパーウェアPASOPIA〔1〕
第1回「PASOPIAプログラム・コンテスト」の優秀作品を集め，システム，ビジネス，教育，ゲームの4編構成．パソピア3機種対応プログラムを満載．
B5判228頁　定価1,600円(〒250)

ペーパーウェア ベーシックマスター〔1〕
'83年度「ベーシックマスタープログラム・コンテスト」の入選作品を一挙掲載．L3，L3MARK5，Jr.ユーザーには必読の珠玉のプログラム集．
B5判148頁　定価1,600円(〒250)

ペーパーウェアPASOPIA〔2〕
P·W PASOPIAの第2弾！第2回「PASOPIAプログラム・コンテスト」の入賞・優秀作品を集め全リストを公開．
B5判164頁　定価1,600円(〒250)

I/O別冊

APPLE and PET 1，2
アップル/ペットのユーザーと6502ファンのためのガイド・ブック
B5判288頁　定価2,500円(〒300)

プログラム電卓ゲーム❶，❷
この本でポケット・コンピュータがゲーム・マシンに変身(!?)．アイデア溢れるプログラム・ゲーム集．
A5判　各定価1,200円(〒250)

プログラム電卓ゲーム❸
ポケコン傑作ゲーム集の第3弾！厳選されたプログラムを，お届けします．
A5判　定価1,500円(〒250)

グラフィック・プリンタの使い方
プリンタをリスト・アウトの為だけに使っていた…そんな方にぜひお薦めの一冊．「ヘェー」と感心すること，ためになること請け合い．
B5判212頁　定価1,900円(〒300)

ポケコン・プログラミング入門
PC-1500の多彩な機能をフル活用するためのバイブル．知らない人には基本から，知ってる人にはとっておきのプログラミング・テクニックを教えます．
A5判204頁　定価1,500円(〒250)

ポケコン活用術
PC-1500の機械語プログラムを作るノウハウ，限界まで使いこなすハイ・テクニックを徹底解説．機械語ビギナーにも最適の入門書です．
A5判296頁　定価1,500円(〒300)

PC-2001プログラム・ライブラリ
ハンドヘルド・コンピュータPC-2001をビジネスからホビーまで多目的に活用できる本です．カセットと併せてご利用ください．
B5判196頁　定価1,900円(〒250)

パソコン計測・制御の実験と製作
パソコンを使った外部制御・自動制御のソフト・ハードがこの一冊でマスター．話題のマイクロマウスなどの実験・製作の実例を詳解．
B5判244頁　定価1,900円(〒300)

WICS・BASE プログラム集
整数型インタープリタ，コンパイラを備えた言語WICS．アセンブラBASE，そのソース・リストを全公開．リアル・タイム高速ゲームも併載．MZユーザー待望の一冊．
B5判336頁　定価2,500円(〒300)

PROGRAM CONTEST作品集 1
「I/Oプログラム・コンテスト」参加作品の中から，特に優秀なものばかり集めた作品集です．
B5判296頁　定価2,500円(〒300)

CPUシリーズ

CPU 1，2，3，4
まったくの入門者にもBASICをわかりやすく解説，ビギナーの気持に立って編集．TVパソコン番組「コンピュートないと」のテキストとしてもご利用いただけます．
B5判　各定価650円(〒250)

CPU 5
特集 パソコン・プログラミング入門
BASICからマシン語までプログラム作りのノウ・ハウを実例入りで解説します．また人工知能，OS-9の話，面白ゲームも満載．
B5判　定価650円(〒250)

徹底研究シリーズ

マイコン徹底研究
M6800をハードからソフトまで初心者にもわかるように，ていねいに解説．マイコンの入門書として大好評！
B5判256頁　定価1,900円(〒300)

BASICゲーム徹底研究
Tiny BASICやレベルⅠBASICのプログラミングの基礎から応用まで，徹底的に解説しました．
B5判264頁　定価1,900円(〒300)

マシン語徹底研究
〝マシン語〟はどうもわからない，という人のための入門書．Z80，8080，6800，6502を解説．
B5判310頁　定価1,900円(〒300)

マイコン・ソフト徹底研究
アセンブラ入門からDOSの作り方までソフトに強くなりたい人には必読の一冊．
B5判304頁　定価1,900円(〒300)

マイコン・ゲーム徹底研究 2
SLOTゲーム，野球ゲームなど60編以上収録．あなたをマイコン・ゲームのエキスパートにするプログラム集．
B5判280頁　定価1,900円(〒300)

マイコン・ゲームの本 1 2，3，4，5
面白マイコン・ゲームがギュウギュウ詰め．ゲーム好きの君なら，この本を持ってなきゃマニアじゃない(!?)．
B5判　各定価1,900円(〒300)

I/O BOOKS

第2種情報処理技術者 試験問題徹底解説〔59年度版〕
第2種情報処理技術者試験問題を徹底解説．受験者のテキストとして格好の書．
赤松　徹著　A5判544頁　定価2,500円(〒300)

CAP-X入門〔59年度版〕
わずか12の命令を覚えるだけで，あなたにもアセンブラがわかる．アセンブラ入門者向き．
赤松　徹著　A5判580頁　定価2,500円(〒300)

PASCAL入門
60もの豊富な例題でPASCALをわかりやすく解説．英・独・米でも出版
I.R.ウィルソン A.M.アディマン共著　A5判200頁　定価1,200円(〒250)

UCSD PASCAL演習
多くのマイコンにインプットされ，いまや標準的PASCALとなったUCSD PASCALの開発者Bowlesの名著．
ケネス.L.ボウルズ著　A5判448頁　定価2,900円(〒300)

BASICサブルーチン・ハンドブック
サブルーチン化すればビギナーでも楽にプログラミング．役立つサブルーチン実例を満載．
K.トラクトン著　A5判160頁　定価1,500円(〒250)

プログラム電卓による土木設計
土木(基礎)の設計に手軽なプログラム電卓(FX-602P)をフルに使うための書．
佐々木昌訓著　B5判240頁 定価2,800円(〒300)

UNIX入門
ミニコンの標準的なOSとして知られるユニックスの入門書．豊富な実行例と実践的な内容で好評を得ている．
R.トーマス/J.イェーツ共著 A5判520頁　定価2,900円(〒300)

HC-20科学技術プログラム集 1，2
ハンドヘルド・コンピュータHC-20を実務，教育，学習などに利用するためのプログラムを100本以上収録．
アイ・ビ・シ編　B5判232頁　各定価2,500円(〒300)

リレーショナル・データーベースの使い方
本書はCP/Mマシンで使えるリレーショナル・データベースGBSを例に挙げ，概念や使い方を具体的に解説します．
B5判　定価1,500円(〒250)

FORTH入門
実用高級言語〝FORTH〟の基礎から応用までを，適例を揚げながらわかりやすく解説しています．
レオ・ブロディー著 原道宏訳　A5判440頁　定価2,500円(〒300)

VisiCalc入門
圧到的普及率を誇る作表型の簡易言語VisiCalc(ビジカルク)．パソコンを初めて使う人でも，使いこなせるよう書かれたVisiCalc入門書の決定版です．
ダグラス・ハーガート著　A5判224頁　定価2,000円(〒250)

6809マシン語入門
2進数，16進数の基礎から始まり6809CPUの仕組を解説しているほか，実戦的プログラミング・テクニックを説明．6809ファン待望のマシン語入門書です．
R.ザックス/W.ラビアック共著 A5判396頁 定価2,500円(〒300)

BASIC入門〜ドラゴンと学ぶプログラミング〜
イラストを豊富に使いBASICを基礎の管磯からわかりやすく，面白く解説．子供から大人まで楽しく学べます．
ロドニー・ザックス著　A5判　近日発売

WordStar入門
ワードスターは代表的な英文ワープロのアプリケーション・ソフトとして広く世界で使われています．本書はワープロの手引き書として，わかり易く解説しています．
アーサー・ネイマン著　A5判　近日発売

C-MOS ICの使い方
C-MOS ICの入門書ですが，デジタルの概念，MOS FETの動作原理など基礎的なものも解説．デジタルIC，応用システムを学ぼう，設計しようという方に格好の書です．
宍倉博久　B5判　近日発売

PC-5000ポータブル・コンピュータ入門
16ビット・ポータブル・コンピュータPC-5000の全貌を凝縮し，表計算簡易ソフトTOOLPLANをわかりやすく解説．
池田純一著　A5判176頁　定価2,000円(〒250)

ソフトウェア・ライブラリ 1 〜 9
ポケット・コンピュータをビジネスに，ゲームに，パーソナル・ユースに使いこなすために欠かせない本．
B5判　各定価2,500円(〒300)

BASICソフトウェア集No. 1，2，3，4
「ポケット・コンピュータ友の会」会員のアイデアあふれるポケコン用プログラムを多数収録．
B5判　各定価1,900円(〒300)

PC-1401入門から応用まで
ポケコンとして，関数電卓として，1台で2役の機能を持つPC-1401の入門から応用までを徹底解説．
A5判256頁　定価1,600円(〒250)

ポータブルコンピュータ ビジネスソフト 〔PC-5000汎用サブルーチン集〕
16ビット・ポータブルコンピュータPC-5000の多方面に利用できるサブルーチンをこの一冊に収録．
B5判288頁　定価2,500円(〒300)

TOOLPLAN™活用集 1，2
PC-5000用 日本語対応 表計算簡易ソフトTOOLPLAN™その使い方から実用例までを満載．
B5判　各定価2,900円(〒300)

活用研究シリーズ

ベーシックマスター活用研究
ベーシックマスターL3/Jr.のすべてを公開．プログラム・コンテスト入選作も同時掲載．
B5判332頁　定価2,500円(〒300)

MZ-80B活用研究
MZ-80Bのハイ・パフォーマンスと柔軟性を活かしたコンパイラ，ビジネス・プログラムなど多数収録．
B5判320頁　定価2,500円(〒300)

MZ-80活用研究
〝クリーン・コンピュータ〟MZ-80．その特性〝クリーン〟を活かしたアイデアが集結．
B5判344頁　定価1,900円(〒300)

PC-8001活用研究
パソコン時代の幕開けを飾ったPC-8001．そのソフト開発に必要な情報を満載．
B5判344頁　定価2,500円(〒300)

PC活用研究(8001/8001mkⅡ/8801)
PC80/mkⅡ/88のシステム，アプリケーションからハードまで幅広い貴重な資料を収録．PCファン座右の書．
B5判320頁　定価2,500円(〒300)

FM-8活用研究
内部解析やリアルタイム処理への応用．FM-8の探究を目指すファンたのための手引書．
B5判354頁　定価2,500円(〒300)

FM-7/8活用研究
FM-7の全回路図，BIOSソース・リストの公開をはじめ貴重な資料やプログラムを満載．ユーザー待望の一冊．
B5判328頁　定価2,500円(〒300)

システム/アプリケーション編

FM-11	マイクロコンピュータ支援教育用システム・ソフト	六田嘉明	4
FM-11	画面編集・製図プログラム	六田嘉明	11
	フルカラーグラフィック画面ハードコピー	六田嘉明, 細川治雄	18
	予測計算プログラム	松波栄治	27
FM-7	ALL RAM DISK BASIC	コンピ	31
FM-11	グラフィック・エディタ	Yes	33
	FILE COPY プログラム	中安博司	63
	TortoiseをBASICで	H.KOGCHI	65
	TortoiseをKで	小松 仁	72
	ディスク・コミュニケーション	ポパイ&ジェリー	79
	KコンパイラV.3.6	COMPAC T & St & Sgn	81
	IODOS 9	Wonder Soft Sng	93
	Kコンパイラ有象無象	COMPAC S	105
	Kコンパイラで文字変数を使う	石井正秀	108
FM-7	URA DOS	DOSのTom	113
FM-7	エディタ・アセンブラ↔BASICコミュニケーション	吉川尚之	121
FM-7	F-BASIC DEBUGGING TOOL	日原洋文	127
FM-7	micro vi	津田伸秀	129
FM-7	Prolog/F V.2	吉川永一	140
FM-7	MUSIC MAKER	COMPAC T	158

FM-11

マイクロコンピュータ
micro computer 支援 教育用 system software
システムソフトウェア

■六田嘉明

1920年代，S. L. Presseによって始められたティーチング・マシンは，機械的な制御機構をもったものでした．しかし，1971年にインテル社から4001マイクロプロセッサ・ファミリが発表されて以来，マイクロコンピュータの発達はめざましいものがあり，昨今ではこれを教育に利用する試みも急激に増加しています．

ところで，マイコンを使ってCAI(Computer Assisted Instruction)を行なう場合，一つはBASICなどの汎用言語を直接プログラムする方法と，なんらかの支援システム・プログラムを使う方法が考えられます．

前者は当然，プログラミングに熟達した教師しか作ることができないのに対し，後者は計算機に関する特別の知識を持たない教師でも，CAIが利用できることを目的としたものです．

一方，CAIには紙芝居のように画面に説明や問題が表示され，生徒の反応に応じて次の画面が自動的に選択されてゆくという，『フレーム・オリエンテッドCAI』があります．この他に，コンピュータ上で現象のシミュレーションを行ない，対話型で現象を理解させようとする『ゲーム・シミュレーション型CAI』や，さらに人工知能化して，生徒が創意をもって参加できたり，自由な質問に応じられたりする『人工知能型CAI』などがあるとされています．

ここでは，フレーム・オリエンテッドCAI用のシステム・ソフトウェアを作成したので，その概要とプログラム・リストを発表させていただきます．なお，本プログラムは元来，FM-11に，筆者の研究室で試作した音声合成装置を接続した，音声応答式CAIシステム支援用のプログラムです．今回は，音声合成装置を使わずに動くように一部仕様変更したものを発表します．音声合成装置は当研究室で新らたに開発したものです．興味のある方は直接，当研究室へお問い合せください．

プログラムの概要

本システムは，大きくわけて，4つのモジュールから成っています(図1)．

❶教材画面作成用サブシステム

これは，スクリーン・エディット方式を用いた，画面作成用プログラムです．画面上に図形や文字を描き，これをディスクに格納するプログラムです．このプログラムは，『画面編集・製図プログラム』のものとまったく同じです．

図1　CAIシステム利用の流れ図

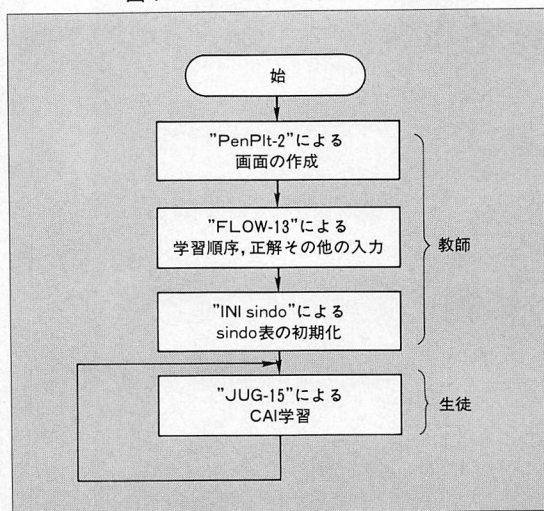



図2　「FLOW-13」走行時の画面例

概要およびプログラム・リストは，そちらを参照してください．

ここで作成した画面情報は，一画面ごとに，ディスクに格納されます．教師は，このプログラムで必要な画面を先ず作成します．

❷授業の流れ設計サブシステム(プログラム名FLOW-13)

❶で作成した画面を，どのような順序で学習させるかを

図3　解答入力時の画面例

図4　ヒントおよび正解表示時の画面

入力するのが，このプログラムです．走行時の画面例を図2に示します．

各画面ごとに，画面表示前後に出力する音声文（音声を出さないときは，何でも良い），ヒントの文，正解，配点，合格点，合格点以上のときの次の先行画面名，合格点以下のときの先行画面名，その他をここで入力します．

入力情報は画面名の後に〝.ATR〞という拡張名が付いてディスクにセーブされます．画面ファイルと〝.ATR〞ファイルとは対をなしており，次に述べるCAI授業実行モジュールによって使われます．

このように，各画面に対して生徒の反応，得点に応じた先行画面名を与えておくことによって，CAIを実行した場合の生徒の反応，得点に応じた画面が自動的に選択されてゆくことになります．このアイディアは，森本氏ら[1]にヒントを得たものです．

❸授業実行サブシステム（プログラム名JUG-15とINIsindo）

これは，❶，❷で作成した教材ファイルを基に，CAI授業を展開してゆくモジュールです．

なお，〝JUG-15〞走行に先立って，一度だけプログラム〝INIsindo〞を走行させておく必要があります．これは，生徒の進度を記録したファイル〝sindo〞を作成し，初期化するためのものです．これをRUNさせると，最も初めに提示する画面の画面名を尋ねるので，これに答えてください．

これで，各生徒（0～49番まで）のsindoには，最初に学習するべき画面の名前が書き込まれます．このsindo表はCAIセッションが終了すると，その生徒が次に学習すべき画面の名前に書き換えられます．

このsindo表の利用により，多人数の生徒の進度を簡単に管理でき，次回からはその生徒の前回の続きを提示することができることになります．

図3～4に，本プログラム走行時の学習進行状況を示しておきます．

なお，本プログラムは現在CAI学会誌に投稿中であることを付記しておきます．

《問い合わせ先》

〒630 奈良県奈良市高畑町　奈良教育大学技術課

参考文献

1) 森本，中山，深川，吉田：〝マイコンを使用した演算機能をもつCAIについて〞，信学技報 Vol. 82, No. 89 (1982) ET82-3, p. 47～52.

リスト1　FLOW-13プログラム・リスト

```
1000 '****************************************
1010 '*FRAME ORIENTED CAI- AUTHORING SYSTEM -  *
1020 '* PART-II(FLOW AND ATR. OF EACH FRAME )  *
1030 '*  CODED BY Y. MUDA,  1983.7.18(OK)      *
1035 '*  RIVESED            1983.9.13(OK)      *
1040 '****************************************
1045 POKE &HFC12, &H17
1050 SCREEN 5,0,0 :'Graphic Scroll MODE,Active P
age=0, Disp Page=0
1060 CLS:CONSOLE0,25,0,0: WIDTH 40,25 : CA=4 : C
B=6 : CC=5
1070 LOCATE 2,2:COLOR 5:PRINT "    ----<< ﾌﾟﾛｸﾞﾗ
ﾑ FLOW >> ----":PRINT
1080 LOCATE 10,5:COLOR 2:KANJI &H3C42,&H3954,&H2
437,&H243F,&H2424,&H4856,&H3966,
&H2472,&H253F,&H2524,&H2557,&H2437,&H2446,&H323C
,&H2435,&H2424,&H2123
1090 LOCATE 10,7:COLOR 6:KANJI &H3358,&H3D2C,&H2
44E,&H4E2E,&H246C,&H244E,&H405F,
&H3757;:PRINT "......(1)" :'ｶﾞｸｼｭｳ ﾉ ﾅｶﾞﾚ ﾉ ｾｯｹｲ
1100 LOCATE 10,9:KANJI&H3358,&H3D2C,&H244E,&H4E2
E,&H246C,&H244E,&H4A51,&H3939;:P
RINT"......(2)": '    ﾍﾝｺｳ
1110 LOCATE 10,11 : KANJI &H3D2A,&H246F,&H246A;:
PRINT "...........(3)"
1120 PRINT : INPUT FLOW
1130 ON FLOW GOTO 1160,1830,1140
1140 END
1150 '******************************
1160 '*  ｶﾞｸｼｭｳﾉ ﾅｶﾞﾚﾉ ｼﾝｷ ﾆｭｳﾘｮｸ   *
1170 '******************************
1175 PRINT
1180 COLOR CA:KANJI &H3268,&H4C4C,&H4C3E;:COLOR
CC:INPUT " ";FILE$:PRINT:' ｶﾞﾒﾝﾒ
ｲ?
1190 COLOR CA:KANJI &H3268,&H4C4C,&H2472,&H3D50,
&H2439,&H4130,&H244E,&H323B,&H40
3C,&H4A38;:COLOR CC:INPUT " ";OSMAE$: PRINT:' ｶﾞ
ﾒﾝ ｦ ﾀﾞｽﾏｴ ﾉ ｵﾝｾｲ ﾌﾞﾝ?
1200 COLOR CA:KANJI &H3268,&H4C4C,&H2472,&H3D50,
&H2437,&H243F,&H3865,&H244E,&H32
3B,&H403C,&H4A38;:COLOR CC:INPUT " ";OSBUN$:PRIN
T:' ｶﾞﾒﾝ ｦ ﾀﾞｼﾀ ｱﾄ ﾉ ｵﾝｾｲ ﾌﾞﾝ
1210 COLOR CA:KANJI &H4C64,&H426A,&H3268,&H4C4C,
&H2447,&H2439,&H242B,&H214A,&H23
31,&H214B,&H2124,&H234D,&H2345,&H234E,&H2355,&H3
268,&H4C4C,&H2447,&H2439,&H242B,
&H214A,&H2330,&H214B;:COLOR CC
1220 INPUT MOME :PRINT:' ﾓﾝﾀﾞｲ ｶﾞﾒﾝ? ﾒﾆｭｰ ｶﾞﾒﾝ?
1230 IF MOME=0 THEN GOTO 1540
1240  '
1250  '   ﾓﾝﾀﾞｲｰｶﾞﾒﾝﾉ ﾊﾞｱｲ
1260  '
1270       COLOR CA:KANJI &H4C64,&H426A,&H244E,&H3
F74,&H214A,&H2330,&H2141,&H2331,
&H2330,&H214B;:COLOR CC:INPUT " ";N:' ﾓﾝﾀﾞｲ ﾉ ｶｽ
ﾞ?
1275 'HARDC 4
1280   GTEN=0
1290    FOR I=0 TO N-1 : PRINT
1300      COLOR CB:PRINT I+1;:KANJI &H4856,&H244E
,&H3272,&H457A;:COLOR CC:INPUT "
 ";ANS$(I):PRINT: '  ...ﾉ ｶｲﾄｳ
1310       COLOR CB: PRINT I+1;:KANJI &H4856,&H24
```

リスト1 FLOW-13プログラム・リスト

```
4E,&H4540,&H3F74;:COLOR CC:INPUT
 " ";TEN(I):PRINT: '  ･･･ﾉ ﾃﾝｽｳ
1320        GTEN=GTEN+TEN(I)
1330 '       COLOR CB:PRINT I+1;:KANJI &H4856,&H2
44E,&H2552,&H2573,&H2548,&H244E,
&H4A38;:COLOR CC:INPUT " ";HBUN$(I):PRINT:' ･･･ﾉ
 ﾋﾝﾄ ﾉ ﾌﾞﾝ
1331         COLOR CB:PRINT I+1;:KANJI &H4856,&H2
44E,&H2552,&H2573,&H2548,&H244E,
&H4A38;:COLOR CC:'IF I=2 THEN HARDC 4
1335         INPUT " ";HBUN$(I): PRINT : ' ･･･ﾉ ﾋ
ﾝﾄ ﾉ ﾌﾞﾝ
1340        COLOR CB : PRINT I+1;:KANJI &H4856,&H
244E,&H2552,&H2573,&H2548,&H244E
,&H323B,&H403C;:COLOR CC:INPUT " ";HOS$(I):PRINT
 :' ･･･ﾉ ﾋﾝﾄ ﾉ ｵﾝｾｲ
1350      NEXT I
1360      '
1370      COLOR CA: KANJI &H3A47,&H4463,&H2569,&H
2524,&H2573,&H244E,&H4540,&H3F74
;:COLOR CC:INPUT " "; STEN: PRINT       :' ｻｲﾃｲ ﾗ
ｲﾝﾉ ﾃﾝｽｳ
1380      COLOR CA:KANJI &H4134,&H4C64,&H4035,&H3
272,&H3B7E,&H244E,&H3954,&H4068,
&H2555,&H2521,&H2524,&H256B,&H4C3E;:COLOR CC:INP
UT " ";SEI$:PRINT:' ｾﾞﾝﾓﾝ ｾｲｶｲ ｼ
ﾞ ﾉ ｲｷｻｷ
1390      COLOR CA:KANJI &H3A47,&H4463,&H2569,&H2
524,&H2573,&H304A,&H3E65,&H244E,
&H3B7E,&H244E,&H3954,&H4068,&H2555,&H2521,&H2524
,&H256B,&H4C3E;:COLOR CC
1400      INPUT" ";SEIM$:PRINT:' ﾀｶｲ ﾃﾝﾉ ﾄｷﾉ ｲｷｻｷ
1410      COLOR CA:KANJI &H3A47,&H4463,&H2569,&H2
524,&H2573,&H4C24,&H4B7E,&H244E,
&H3B7E,&H244E,&H3954,&H4068,&H2555,&H2521,&H2524
,&H256B,&H4C3E;:COLOR CC
1420      INPUT" ";MACHI$:PRINT:' ﾋｸｲﾃﾝﾉ ﾄｷﾉ ｲｷｻｷ
1430      COLOR CA:KANJI &H386D,&H457A,&H244E,&H3
E6C,&H3967,&H2124,&H4631,&H2438,
&H4C64,&H426A,&H2472,&H242F,&H246A,&H242B,&H2428
,&H2437,&H245E,&H2439,&H242B,&H2
14A,&H2161,&H2331,&H214B,&H2124;
1440      KANJI &H3C21,&H244E,&H4C64,&H426A,&H245
8,&H3F4A,&H245F,&H245E,&H2439,&H
242B,&H214A,&H2161,&H2330,&H214B,&H2123;:COLOR C
C:INPUT" ";AGAIN:PRINT:' ｺﾞﾄｳﾉ ﾄ
ｷ ﾂｷﾞﾍ ｽｽﾑｶ? ｸﾘｶｴｽｶ?
1450      COLOR CA:KANJI &H386D,&H246A,&H244E,&H4
247,&H2441,&H405A,&H246A,&H3273,
&H3F74,&H244F,&H214A,&H2330,&H2141,&H2339,&H2339
,&H214B;:COLOR CC
1460      INPUT" ";TIMS:PRINT: ' ｱﾔﾏﾘ ﾉ ｳﾁｷﾘ ｶｲｽｳ
?
1470      COLOR CA:KANJI &H412A,&H4272,&H3B5E,&H2
472,&H467E,&H246C,&H245E,&H2349,
&H242B,&H2123;:PRINT" (Yes=1, No=0) ";: COLOR CC
1480      INPUT " ";YNSEN:PRINT  :'ｾﾝﾀｸｼ ｲﾚﾙｶ?
1490      ST$="dummy":IF YNSEN=0THEN GOTO 1510
1500       COLOR CA:KANJI &H412A,&H4272,&H3B5E,&H
244E,&H3F74,&H244F,&H214A,&H2522
,&H2141,&H2531,&H214B;:COLOR CC:INPUT " ｱ-"; ST$
: PRINT   :' ｾﾝﾀｸ ｼ ﾉ ｶｽﾞ?(ｱ-ｹ)
1510  '   GOSUB 1660 : GOTO 1060
1515      GOSUB 1660 :'HARDC 4
1518      END
1520  '
1530  ' MENU ｶﾞﾒﾝﾉ ﾊﾞｱｲ
1540  '
1550      YNSEN=1
1560      COLOR CA:KANJI &H412A,&H4272,&H3B5E,&H2
44E,&H3F74,&H244F,&H214A,&H252,&
H2141,&H2531,&H214B;:COLOR CC:'ｾﾝﾀｸｼﾉ ｶｽﾞﾊ(ｱ-ｹ)
1570      INPUT"  ｱ-";ST$ :PRINT:IF ST$="" THEN G
OTO 1640
1580      COLOR CA:KANJI &H412A,&H4272,&H3B5E,&H2
44E,&H3954,&H4068:PRINT:' ｾﾝﾀｸｼ
ﾉ ｲｷｻｷ?
1590      BB=ASC(ST$)
1600      FOR I=&HB1 TO BB
1610        COLOR CB:PRINT CHR$(I);:KANJI &H244E,
&H3954,&H4068,&H2555,&H2521,&H25
24,&H256B,&H4C3E;:COLOR CC:INPUT " "; SENG$(I-&H
B1)
1620        PRINT
1630      NEXT I
1640      GOSUB 1660 : GOTO 1060
1650 '
1660 ' FILEWR
1670      ON ERROR GOTO 2740
1680      A$=FILE$+".ATR"
1690      OPEN "O",#1,A$
1700      WRITE #1,OSMAE$,OSBUN$,YNSEN,ST$,MOME,N
,GTEN,STEN,SEI$,SEIM$,MACHI$,AGA
IN,TIMS,NUM
1710      IF MOME=0 THEN GOTO 1760
1720        FOR I=0 TO N-1
1730          WRITE #1,ANS$(I),TEN(I),HBUN$(I),HO
S$(I)
1740        NEXT I
1750        GOTO 1800 :'CLOSE
1760       '*SENSID
1770        FOR I=&HB1 TO BB
1780          WRITE #1,SENG$(I-&HB1)
1790        NEXT I
1800      CLOSE #1: ON ERROR GOTO 0
1810      RETURN
1820 '***************************
1830 '*  ｼﾞｭｷﾞｮｳﾉ ﾅｶﾞﾚﾉ ﾍﾝｺｳ    *
1840 '***************************
1845 PRINT
1850 COLOR CA:KANJI &H3268,&H4C4C,&H4C3E;:COLOR
CC:INPUT " "; FILE$:' ｶﾞﾒﾝ ﾒｲ?
1860 GOSUB 2550 : PRINT :'*FILERD
1870 COLOR CA:KANJI &H3268,&H4C4C,&H2472,&H3D50,
&H2439,&H4130,&H244E,&H323B,&H40
3C,&H4A38:' ｶﾞﾒﾝ ｦ ﾀﾞｽﾏｴ ﾉ ｵﾝｾｲ ﾌﾞﾝ
1880 AA$=OSMAE$:GOSUB 2460:OSMAE$=AA$
1890 COLOR CA:KANJI &H3268,&H4C4C,&H2472,&H3D50,
&H2437,&H243F,&H3865,&H244E,&H32
3B,&H403C,&H4A38:' ｶﾞﾒﾝ ｦ ﾀﾞｼﾀ ｱﾄ ﾉ ｵﾝｾｲ ﾌﾞﾝ
1900 AA$=OSBUN$ : GOSUB 2460 : OSBUN$=AA$
1910 IF MOME=0 THEN GOTO 2330
1920   '
1930   '   ﾓﾝﾀﾞｲｰｶﾞﾒﾝﾉ ﾊﾞｱｲ
1940   '
1950    COLOR CA:KANJI &H4C64,&H426A,&H244E,&H3F7
4,&H214A,&H2330,&H2141,&H2331,&H
2330,&H214B:' ﾓﾝﾀﾞｲ ﾉ ｶｽﾞ
1960    COLOR CB:PRINT N;:COLOR CC:INPUT TEMP$ :
TEMP=VAL(TEMP$)
1970    IF TEMP<>0 THEN N=TEMP
1980    PRINT N : PRINT
1990    FOR I=0 TO N-1 : PRINT
2000      COLOR CA:PRINT I+1;:KANJI &H4856,&H244E
,&H3272,&H457A:'･･･ﾉ ｶｲﾄｳ
2010      AA$=ANS$(I) : GOSUB 2460 : ANS$(I)=AA$
2020      COLOR CA:PRINT I+1;:KANJI &H4856,&H244E
,&H2552,&H2573,&H2548,&H244E,&H4
A38:'･･･ﾉ ﾋﾝﾄ ﾉ ﾌﾞﾝ
2030      AA$=HBUN$(I):GOSUB 2460:HBUN$(I)=AA$
2040      COLOR CA:PRINT I+1;:KANJI &H4856,&H244E
,&H2552,&H2573,&H2548,&H244E,&H3
23B,&H403C:'･･･ﾉ ﾋﾝﾄ ﾉ ｵﾝｾｲ
2050      AA$=HOS$(I):GOSUB 2460:HOS$(I)=AA$
2060      COLOR CA:PRINT I+1;:KANJI &H4856,&H244E
,&H4540,&H3F74:' ･･･ﾉ ﾃﾝｽｳ
2070      COLOR CB : PRINT TEN(I);: COLOR CC :

          INPUT TEMP$: TEMP=VAL(TEMP$)
2080      IF TEMP<>0 THEN TEN(I)=TEMP
2090      PRINT TEN(I)
2100    NEXT I: PRINT
2110    COLOR CA:KANJI &H3A47,&H4463,&H2569,&H252
4,&H2573,&H244E,&H4540,&H3F74:'
ｻｲﾃｲ ﾃﾝ
2120    COLOR CB : PRINT STEN;: COLOR CC :

        INPUT TEMP$: TEMP=VAL(TEMP$)
2130    IF TEMP<>0 THEN STEN=TEMP
2140    PRINT STEN
2150    COLOR CA:KANJI &H4134,&H4C64,&H4035,&H327
2,&H3B7E,&H244E,&H3954,&H4068,&H
2555,&H2521,&H2524,&H256B,&H4C3E:' ｾﾞﾝﾓﾝ ｾｲｶｲ ｼﾞ
2160    AA$=SEI$ : GOSUB 2460 : SEI$=AA$
2170    COLOR CA:KANJI &H3A47,&H4463,&H2569,&H252
4,&H2573,&H304A,&H3E65,&H244E,&H
3B7E,&H244E,&H3954,&H4068,&H2555,&H2521,&H2524,&
H256B,&H4C3E:' ﾀｶｲ ﾃﾝ ﾉ ﾄｷ
2180    AA$=SEIM$ : GOSUB 2460 : SEIM$=AA$
2190    COLOR CA:KANJI &H3A47,&H4463,&H2569,&H252
4,&H2573,&H4C24,&H4B7E,&H244E,&H
3B7E,&H244E,&H3954,&H4068,&H2555,&H2521,&H2524,&
H256B,&H4C3E:' ﾋｸｲﾃﾝ ﾉ ﾄｷ
2200    AA$=MACHI$ : GOSUB 2460 : MACHI$=AA$
2210    COLOR CA:KANJI &H386D,&H246A,&H244E,&H424
7,&H2441,&H405A,&H246A,&H3273,&H
3F74,&H244F,&H214A,&H2330,&H2141,&H2339,&H2339,&
H214B:' ｱﾔﾏﾘﾉ ｳﾁｷﾘ ｶｲｽｳ
```

リスト1　FLOW-13プログラム・リスト

```
2220    COLOR CB : PRINT TIMS;: COLOR CC :

        INPUT TEMP$
2230    IF TEMP$=""THEN 2240 ELSE TIMS=VAL(TEMP$)
2240    PRINT TIMS
2250     COLOR CA:KANJI &H412A,&H4272,&H3B5E,&H24
72,&H467E,&H246C,&H245E,&H2349,&
H242B,&H2123;:PRINT"(Y,y,N,n or ret)";:COLOR CC
2260    COLOR CB: PRINT YNSEN;: COLOR CC:

        INPUT TEMP$:IF TEMP$=""THEN 2280:'ｾﾝﾀｸｼﾗ
ｲﾚﾙｶ?
2270    IF TEMP$="Y" OR TEMP$="y" THEN YNSEN=1 EL
SE YNSEN=0
2280    PRINT YNSEN
2290    IF YNSEN=0 THEN 2440
2300 '
2310 ' MENU ｶﾞﾒﾝﾉ ﾊﾞｱｲ
2320 '
2330    COLOR CA:KANJI &H412A,&H4272,&H3B5E,&H244
E,&H3F74,&H244F,&H214A,&H2522,&H
2141,&H2531,&H214B:' ｾﾝﾀｸ ｼ ﾉ ｶｽﾞ (ｱｰｹ)
2340    COLOR CB:PRINT "ｱｰ";ST$;:COLOR CC:INPUT K
ST$
2350    IF KST$<>"" THEN ST$=KST$
2360    PRINT ST$
2370    IF MOME=1 OR ST$="" THEN GOTO 2440
2380      COLOR CA:KANJI &H412A,&H4272,&H3B5E,&H2
44E,&H3954,&H4068:' ｶｸｾﾝﾀｸｼﾉ ｲｷｻ
ｷ
2390      BB=ASC(ST$)
2400    FOR I=&HB1 TO BB
2410        COLOR CA:PRINT CHR$(I);:KANJI &H244E,
&H3954,&H4068,&H2555,&H2521,&H25
24,&H256B,&H4C3E
2420        AA$=SENG$(I-&HB1):GOSUB 2460:SENG$(I-
&HB1)=AA$
2430      NEXT I
2440    GOSUB 1660 : GOTO 1060
2450 '
2460 '*EDIT-INPUT AA$
2470 '
2480     COLOR CB : PRINT AA$
2490     COLOR CC : LINE INPUT AB$
2500     IF AB$<>"" THEN AA$=AB$
2510     PRINT AA$: PRINT
2530     RETURN
2540 '
2550 '*FILERD
2560 '
2570 ON ERROR GOTO 2750
2580     A$=FILE$+".ATR"
2590     OPEN "I",#1,A$
2600     INPUT #1,OSMAE$,OSBUN$,YNSEN,ST$,MOME,N,
GTEN,STEN,SEI$,SEIM$,MACHI$,AGAI
N,TIMS,NUM
2610     IF MOME=0 THEN GOTO 2660
2620      FOR I=0 TO N-1
2630        INPUT#1,ANS$(I),TEN(I),HBUN$(I),HOS$(I
)
2640      NEXT I
2650      GOTO 2710
2660     'MENU ｶﾞﾒﾝ ﾉ ﾊﾞｱｲ
2670     BB=ASC(ST$)
2680     FOR I=&HB1 TO BB
2690       INPUT #1,SENG$(I-&HB1)
2700     NEXT I
2710     CLOSE #1 :ON ERROR GOTO 0
2720     RETURN
2730 'ERROR ｼｮﾘ
2740 IF ERR=64 THEN KILL A$: RESUME 1690:'*FILE-
ALREADY EXIST ERROR
2750 RESUME NEXT
2760 ' SAVE"FLOW-13"
2770 ' 1275, 1331, 1335, 1515 ﾆ HARDC 4 ｱﾘ。
```

リスト2　INI sindoプログラム・リスト

```
100 '******************************
110 '*    INITIALIZE "sindo"       *
120 '******************************
130 OPEN "O", #1, "sindo"
140 INPUT "ﾓｯﾄﾓ ｻｲｼｮﾆ ﾋｮｳｼﾞｽﾙ FRAME(ｶﾞﾒﾝ)ﾒｲ ﾊ (d
efault=PO) ";TFNM$
150 IF TFNM$="" THEN TFNM$="PO"
160 FOR II=0 TO 49
170     ! #1, TFNM$
180 NEXT II
190 CLOSE #1
200 END
210 'SAVE "INIsindo"
```

リスト3　JUG-15プログラム・リスト

```
100 '****************************************
110 '* FRAME ORIENTED CAI - EXCUTION MODELE -  *
120 '*             CODED BY Y. MUDA, 1983.7.21  *
125 '*             RIVESED           1983.9.14  *
130 '****************************************
135  POKE &HFC12,&H17:'400 dots mode HARDC ﾖｳ
140  ON COM (0) GOSUB 4360 :'ｵﾝｾｲ ｺﾞｳｾｲ ｿｳﾁｦ ﾂﾅｸ
ﾞﾊﾞｱｲﾆ ﾋﾂﾖｳ。
150  CONSOLE 0,25,0: WIDTH 80,25 : COLOR 7,0,7,0
 : CLS
160  DEFINT A-Z : DEFSNG T,R
170  LSENTK=19 : CMN=33: ALIM=200: BLIM=100: CLI
M=25: DLIM=1700: GLIM=300
180  DIM GP0(GLIM),GP1(GLIM),GP2(GLIM),GP3(GLIM)
,GP4(GLIM)
190  DIM GP5(GLIM),GP6(GLIM),GP7(GLIM),GP8(GLIM)
,GP9(GLIM)
200  DIM CMMAND%(CMN),SD$(50)
210  '
220  DEF FNKAN$(KC)=CHR$(KC ¥ &H100)+CHR$(KC MOD
 &H100)
230  RESTORE 250
240  FOR II=1 TO CMN : READ CMMAND%(II) : NEXT I
I
250 '*Command LIST
260 DATA32,&H73,48,&H6C,&H62,&H63,28,29,30,31
270  '     s,   0,  l,   b,   c,  ->,<-,▲,▼
280 DATA49,50,51,52,53,54,55,56,57,&H77
290  '1  2  3  4  5  6  7  8  9  w
300 DATA&H70,&H6D,&H67,&H78,&H6B,&H61,&H2B,&H72
310  '   p    m    g    x    k    a    +    r
320  DATA &H69,&H2E,&H64,&H6F,&H2F
330  '    i    .    d    o    /
340  '
350  GPN$="SGP" : GOSUB 3280 :'*RDGP
360  OPEN "I",#1,"sindo"
370    FOR II=0 TO 49:INPUT #1,SD$(II):NEXTII
380    CLOSE #1
390  E=0:TTL=0:GTOTL=0:'E=CNTL.VAR.FOR SKIPask(c
ont or not)
400  LOCATE 16,10:KANJI &H2422,&H244A,&H243F,&H2
44E,&H4856,&H3966,&H2472,&H253F,
&H2524,&H2557,&H2437,&H2446,&H323C,&H2435,&H2424
,&H2123;:INPUT NO
410                    ' ｱﾅﾀﾉ ﾊﾞﾝｺﾞｳ ｦ ﾀｲﾌﾟ ｼﾃｸﾀﾞｻｲ
420  B$=SD$(NO-1)
430 '*********************************
440 '*          NEW FRAME             *
450 '*********************************
460  ON ERROR GOTO 4510
470  CLS:IF B$="EEE"THEN GOTO 1260:'COURSE ENDﾍ
480  GOSUB 1980 :'FILER
490  E=E+1
500  IF E=1 THEN GOTO 530 :' ﾊｼﾞﾒ
510   CONSOLE 0,24:LOCATE 20,8:KANJI &H3358,&H3D
2C,&H2472,&H4233,&H2431,&H245E,&
H2439,&H242B;:'"ｶﾞｸｼｭｳ ｦ ﾂﾂﾞｹﾏｽｶ
515     AA$="ｶﾞｸｼｭｳ ｦ ﾂﾂﾞｹﾏｽｶ ":GOSUB 3560:'*HASS
EI
518     INPUT " (Yes=1,No=0)"; CTN
519 'HARDC 4
520     IF CTN=0THEN GOTO 1310:'ﾎﾝｼﾞﾂ ｼｭｳﾘｮｳ ﾍ
530    ' ﾊｼﾞﾒ
540     AA$=OSMAE$:GOSUB 3560:'*HASSEI
550     GOSUB 2180 :'*DRAWF
560     AA$=OSBUN$ :GOSUB 3560:'*HASSEI
570     BB=ASC(ST$)
580     IF YNSEN=1THEN GOSUB 1470:'ｾﾝﾀｸｼ ﾋｮｳｼﾞ
590     IF MOME=1THEN GOTO 750:'ﾓﾝﾀﾞｲ-FRAME ﾉ ﾊﾞｱ
ｲ
600     '
610     ' MENU-FRAME ﾉ ﾊﾞｱｲ
620     '
630     LOCATE 16,LSENTK+2:INPUT B$:B$=SENG$(VAL(
B$)-VAL("ｱ")):GOTO 1210:' NEXT-F
ILE
640     '
650     PEN ON : ON PEN GOSUB 670 :' PENSUB
660     GOTO 660
670     'PENSUB
```

リスト3　JUG-15プログラム・リスト

```
680     PEN OFF : BEEP
690     PX=PEN(1)-4:PY=PEN(2):IF(PY<LSENTK) OR (P
Y>LSENTK+1) THEN GOSUB 1670:RETU
RN 650:' ﾜｸｶﾞｲ
700     FOR II=0 TO BB-&HB1
710       IF PX<=(4*(II+1)+12) THEN B$=SENG$(II):
RETURN 1210:' NEXT-FILE
720     NEXT II
730     GOSUB 1670 : RETURN 650 :' ﾜｸｶﾞｲ
740     '
750     ' ﾓﾝﾀﾞｲ-FRAME ﾉ ﾊﾞｱｲ
760     '
770     TENSU=0
780     FOR I=0 TO N-1
790       TIMSA=0
800       ' ﾓｰｲﾁﾄﾞ
810       LOCATE 2,LSENTK+2:COLOR 7:PRINT I+1;:
KANJI &H4856,&H244E,&H457A,&H212
9;:' ﾊﾞﾝ ﾉ ｺﾀｴ
820       LOCATE 0,0:PRINT TIMSA+1;:KANJI &H327
3,&H2461:'ｶｲﾒ
830       IF I<9 THEN AA$=CHR$(I+&H31) ELSE AA$
="1"+CHR$(I-10+&H31)
840       AA$=AA$+"ﾊﾞﾝﾉ"+CHR$(TIMSA+&H31)+"ｶｲﾒ"
 : GOSUB 3560 :'*HASSEI
850       IF YNSEN=1 THEN GOSUB 1540 ELSE LOCAT
E 16,LSENTK+2:INPUT ANSA$ :'*GET
AFLP
860       GOSUB 1730 :'*HANTEI
870       IF SEI=1 THEN GOTO 1040 :'*SEIKAI
880       '
890       ' ｱﾔﾏﾘ
900       '
910       BEEP:LOCATE 26,0:AA$="ﾏﾁｶﾞｲﾃﾞｽ｡":KANJ
I &H3456,&H3063,&H2424,&H2447,&H
2439,&H2123:GOSUB 3560:'*HASSEI
920       GOSUB 4400 :'*SLEEP
930       LOCATE 26,0:PRINT "          "
940       LOCATE 2,LSENTK+3:PRINT" ﾋﾝﾄ: ";:PRIN
T HBUN$(I)
950       IF TIMSA=0 THEN AA$="ﾋﾝﾄ  "+HOS$(I):G
OSUB 3560:'*HASSEI
960       TIMSA=TIMSA+1
965 'HARDC 4
970     IF TIMS>TIMSA THEN 800 :' ﾓｰｲﾁﾄﾞ
980       '
990       LOCATE 2,LSENTK+4:KANJI &H4035,&H3272
,&H244F,&H2156;:PRINT ANS$(I);:K
ANJI &H2157,&H2447,&H2439,&H2123,&H2539,&H255A,&
H213C,&H2539,&H252D,&H213C,&H247
2,&H3221,&H2437,&H2446,&H323C,&H2435,&H2424,&H21
23
1000              '*ｾｲｶｲﾊ ･･･ ﾃﾞｽ｡ Space Key ｦ ｵｼﾃ
ｸﾀﾞｻｲ｡
1010        AA$="ｾｲｶｲﾜ"+ANS$(I)+"ﾃﾞｽ｡ ｽﾍﾟｴｽｷｲｦ ｵ
ｼﾃｸﾀﾞｻｲ":GOSUB 3560:GOSUB 4420:'
*HASSEI,   *SLEEP2
1015 'HARDC 4
1020      IF INKEY$="" THEN 1020 ELSE 1100 :' ﾂｷ
ﾞ ﾉ ﾓﾝﾀﾞｲ
1030       '
1040       ' ｾｲｶｲ
1050       '
1060       LOCATE26,0:AA$="ｾｲｶｲﾃﾞｽ｡":KANJI&H403
5,&H3272,&H2447,&H2439,&H2123,&H
2420 : GOSUB 3560 :'*HASSEI
1065       GOSUB 4400 :'*Sleep
1070       LOCATE26,0 : PRINT "
"
1090       TENSU=TENSU+TEN(I)
1100       ' ﾂｷﾞ ﾉ ﾓﾝﾀﾞｲ
1102 'HARDC 4
1105       LINE (0,0)-(639,30),PRESET,0,BF
1110       LINE (0,LSENTK+2 )-(79,LSENTK+4 ),"
",0,BF
1120       LINE (4,16*(LSENTK+2))-(634,16*(LSEN
TK+5)),PRESET,0,BF
1130     NEXT I
1140     '
1150     ' 1 FRAME ｵﾜﾘ
1160     '
1170     TTL=TTL+TENSU : GTOTL=GTOTL+GTEN
1180     IF TENSU=GTEN THEN B$=SEI$
1190     IF TENSU>=STEN THEN B$=SEIM$ ELSE B$=MACH
I$
1200 '
1210 ' ﾂｷﾞ ﾉ FRAME ﾍ
1220 '
1230     SD$(NO-1)=B$
1240     GOTO 440 :' NEW FRAME ﾍ
1250 '
1260 ' COURSE END
1270 '
1280     CONSOLE 0,24,1 : LOCATE 2, 10 :
1290     PRINT "**** ";:KANJI &H2533,&H213C,&H2539
,&H244F,&H2439,&H2459,&H2446,&H3
D2A,&H246F,&H246A,&H245E,&H2437,&H243F,&H212A,&H
212A;:PRINT " ****":GOTO 1370:'*
* ｺｰｽﾊ ｽﾍﾞﾃｵﾜﾘﾏｼﾀ!! ** :' ｱﾄ ｼｮﾘ ﾍ
1300 '
1310 ' ﾎﾝｼﾞﾂ ｼｭｳﾘｮｳ
1320 '
1330     LOCATE 8,10
1340     KANJI &H3A23,&H467C,&H244E,&H2422,&H244A,
&H243F,&H244E,&H4640,&H4540,&H24
4F;:PRINT " ";TTL;"/";GTOTL;" ";:KANJI &H2447,&H
2437,&H243F,&H2123,&H3C21,&H2462
,&H242C,&H2473,&H2450,&H246A,&H245E,&H2437,&H246
7,&H2426,&H212A
1345       ' ｷｮｳﾉ ｱﾅﾀﾉ ﾄｸﾃﾝﾊ ../.. ﾃﾞｼﾀ｡ ﾂｷﾞﾓ ｶ
ﾞﾝﾊﾞﾘﾏｼｮｳ｡
1350       AA$="ｷｮｳﾉ ｱﾅﾀﾉ ﾄｸﾃﾝﾊ ": GOSUB 3560: AA$=S
TR$(GTOTL)+" ﾃﾝﾏﾝﾃﾝﾉ ": GOSUB 35
60: AA$=STR$(TTL)+" ﾃﾝ ﾃﾞｼﾀ｡ ﾂｷﾞﾓ ｶﾞﾝﾊﾞｯﾃｸﾀﾞｻｲ｡
": GOSUB 3560: '*HASSEI
1360 '
1370 ' ｱﾄ ｼｮﾘ
1380 '
1390    KILL "sindo"
1400    OPEN "O",#2,"sindo"
1410    FOR II=0 TO 49: WRITE #2, SD$(II): NEXT II
1415 'HARDC 4
1420    END
1430 '
1440 '
1450 '
1460 '*****************
1470 '*  ｾﾝﾀｸｼ ﾉ ﾋｮｳｼﾞ  *
1480 '*****************
1490      X=1
1500      FOR II=0 TO BB-&HB1
1510      LOCATE 4*X+10 , LSENTK: COLOR 10: IF II<
5 THEN KANJI II+II+&H2522; ELSE
KANJI II+II+&H2521;
1520      X=X+1: NEXT II: COLOR 7: RETURN
1530 '*****************************************
1540 '*  GETAFLP  (GET ANS$ FROM LIGHT PEN)   *
1550 '*****************************************
1560      LOCATE 16, LSENTK+2 : INPUT ANSA$ : RETU
RN
1570      '
1580      PEN ON: ON PEN GOSUB 1610: '*PENSUBA
1590      GOTO 1590
1600      RETURN
1610 '*PENSUBA
1620      PEN OFF: BEEP
1630      PX=PEN(1)-4 : PY=PEN(2):IF PY < LSENTK O
R PY > LSENTK+1 THEN GOSUB 1670:
 RETURN 1580: ' ﾜｸｶﾞｲ
1640      FOR IP=0 TO BB-&HB1
1650      IF PX<= 4*(IP+1)+12 THEN ANSA$=CHR$(IP+&
HB1): LOCATE 18,LSENTK+2:PRINT A
NSA$: RETURN 1600
1660      NEXT IP : GOSUB 1680: RETURN 1580
1670 ' ﾜｸｶﾞｲ
1680      BEEP : LOCATE 26,0 : PRINT "ﾜｸｶﾗ ﾊｽﾞﾚﾃ ｲ
ﾏｽ｡ ｵｼ ﾅｵｼﾃ ｸﾀﾞｻｲ｡"
1690      FOR II=0 TO 1000 : NEXT II
1700      LOCATE 26, 0 : PRINT "
                                 "
1710      RETURN
1720 '*****************
1730 '*  ｾｲ/ｺﾞ ﾉ ﾊﾝﾃｲ  *
1740 '*****************
1750      AA$=ANSA$ : GOSUB 1890: ANSA$=AB$ : '*DE
LSP
1760      AA$=ANS$(I) : GOSUB 1890 : ANS$(I)=AB$ :
 '*DELSP
1770      NOCHR= LEN(ANS$(I))
1780      FOR JJ=1 TO 5: ANSD$(JJ)="": NEXT JJ: JJ
=1
1790      FOR II=1 TO NOCHR
1800        CC$= MID$( ANS$(I), II, 1)
1810        IF CC$ <> "," THEN ANSD$(JJ)=ANSD$(JJ)
+CC$ ELSE JJ=JJ+1
1820      NEXT II
1830      SEI=0
1840      FOR II=1 TO JJ
```

```
1850        IF ANSA$=ANSD$(II) THEN SEI=1
1860      NEXT II
1870      RETURN
1880 '***************************************
***************
1890 '*  DELSP ( DELETE SPACES IN AA$ AND RETURN
 IT IN AB$ )  *
1900 '***************************************
***************
1910      NOCHR= LEN(AA$): AB$=""
1920      FOR II=1 TO NOCHR
1930        CC$= MID$(AA$, II, 1)
1940        IF CC$<>" " THEN AB$=AB$+CC$
1950      NEXT II
1960      RETURN
1970 '***************
1980 '*  FILE READ  *
1990 '***************
2000      AA$=B$+".ATR"
2010      OPEN "I", #2, AA$
2020      INPUT #2,OSMAE$,OSBUN$,YNSEN,ST$,MOME,N,
GTEN,STEN,SEI$,SEIM$,MACHI$,AGAI
N,TIMS,NUM
2030      IF MOME=0 THEN 2090 :' MENU-ｶﾞﾒﾝ ﾉ ﾊﾞｱｲ
ﾍ
2040      FOR II=0 TO N-1
2050        INPUT #2, ANS$(II),TEN(II),HBUN$(II),H
OS$(II)
2060      NEXT II
2070      CLOSE #2 : RETURN
2080  ' MENU-ｶﾞﾒﾝ ﾉ ﾊﾞｱｲ
2090      BB=ASC(ST$)
2100      FOR II=&HB1 TO BB
2110        INPUT #2, SENG$(II-&HB1)
2120      NEXT II
2130      CLOSE #2
2140      RETURN
2150 '***************************************
******
2160 '*  DRAWF (CAI ﾉ 1 FRAME ｶﾞﾒﾝ ｦ ｽｸﾘｰﾝ ﾆ ｴｶﾞ
ｸ )  *
2170 '***************************************
******
2180      GOTO 2220
2190  ' LPSET
2200      LLPX=LPX : LLPY=LPY : LPX=X : LPY=Y : RE
TURN
2210 '
2220      DIM A(ALIM),TA(BLIM),A$(CLIM),B(DLIM),PI
CTURE%(GLIM)
2230      I=0:J=0:K=0:L=0:C=7:KPC=7:KBC=0
2240      INC0=16:INC1=16:X=19:Y=48:GOSUB 2200:GOS
UB 2200:'*LPSET
2250      CLS: GOSUB 2450 :' ﾜｸ ｦ ｶｸ
2260      OPEN "I",#2,B$
2270      INPUT #2,LIM,LIB,LIC,LID
2280        FOR II=0 TO LIM: INPUT #2, A (II): NEX
T II
2290        FOR II=0 TO LIB: INPUT #2, TA(II): NEX
T II
2300        FOR II=0 TO LIC: INPUT #2, A$(II): NEX
T II
2310        FOR II=0 TO LID: INPUT #2, B (II): NEX
T II
2320        CLOSE #2
2330  '
2340      C=7 : COLOR 7,0,7
2350      GOSUB 2480 : '*GETKEYD
2360      WHILE CH<>1
2370        FOR II=1 TO CMN
2380          IF CMMAND%(II)=CH THEN F=II :GOTO 24
00 ELSE F=-1
2390        NEXT II
2400        ON F GOSUB 2500,3520,2560,2580,2620,26
70,3520,3520,3520,3520,3520,3520
,3520,3520,2550,3520,3520,3520,3520,3520,2710,27
70,2810,3520,3460,2850,3520,2890
,3520,2520, 3520,3150,3520
2402      's,0,l,b,c,->,<-,▲,▼,1,2,3,4,5,6,7,8,9
,w,p,m,g,x,k,a,+,r,i,.,d,o,/
2410        GOSUB 2480 :'*GETKEYD
2420      WEND
2430      I=I-1 : ERASE A,TA,A$,B,PICTURE%: RETURN
2440 '*ﾜｸ ｦ ｶｸ
2450        CONSOLE 0,25,0: CLS 4
2460          LINE (3,32)-(637,397), PSET,7,B:LINE(
3,288)-(637,288),PSET,7
2470        CONSOLE 0,2,0: RETURN
2480 '*GETKEYD
2490      CH=A(I) : I=I+1 : RETURN
2500 '*COLORD
2510      C=B(L) : L=L+1 : COLOR C,0,C : RETURN
2520 '*PSETD
2530      GOSUB 3040 : PSET (X,Y,C):'*GETXY
2540 '*PSETJ
2550      GOTO 2200 :'*LPSET
2560 '*PRESETD
2570      GOSUB 3040 : PRESET (X,Y,C): GOTO 2200 :'
*GETXY,*LPSET
2580 '*LINED
2590      GOSUB 3040 : GOSUB 3060 :'*GETXY,*GETLPXY
2600 '*INEJ
2610      LINE(LPX,LPY)-(X,Y),PSET, C: GOTO 2200 :'
*LPSET
2620 '*BOXD
2630      GOSUB 3040 : GOSUB 3060 :'*GETXY,*GETLPXY
2640 '*BOXJ
2650      LINE(LPX,LPY)-(X,Y),PSET,C,B: GOTO 2200 :
'*LPSET
2660 '*CIRCLED
2670      RADIUS=TA(J): MFCIRC!=TA(J+1): J=J+2: GOS
UB 3040 :'*GETXY
2680 '*CIRCLEJ
2690      CIRCLE(X,Y),RADIUS,C,MFCIRC!: GOTO 2200 :
'*LPSET
2700 '*PAINTD
2710      GOSUB 3100: GOSUB 3040 :'*GETPCBC,*GETXY
2720 '*PAINTJ
2730      FOR II=NBC+1 TO 6: BC(II)=BC(NBC): NEXT I
I :PRESET (X,Y)
2740      PAINT (X,Y),PC,BC(0),BC(1),BC(2),BC(3),BC
(4),BC(5),BC(6)
2750      GOTO 2200: '*LPSET
2760 '*PUTD
2770      GOSUB 3040 :GOSUB 3090 :'*GETXY,*GETDXY
2780 '*PUTJ
2790      PUT@A (X,Y)-(X+DX,Y+DY),PICTURE%,PSET : G
OTO 2200 :'*LPSET
2800 '*GETD
2810      GOSUB 3040: GOSUB 3090 :'*GETXY,*GETDXY
2820 '*GETRJ
2830      GET@A (X,Y)-(X+DX,Y+DY),PICTURE%,G: GOTO
2200 :'*LPSET
2840 '*ARCD
2850      XC=B(L) : YC=B(L+1) : L=L+2
2860      R!=TA(J) : TMIN=TA(J+1) : TMAX=TA(J+2) :
J=J+3
2870      CIRCLE(XC,YC),R!,C,MFCIRC!,TMIN,TMAX
2880      GOTO 2200
2890 '*ALPHD
2900      C$=A$(K): K=K+1: GOSUB 3140: GOSUB 3040:
'*GETMXY,*GETXY
2910 '*ALPHJ
2920      SYMBOL (X,Y),C$,MX,MY,C : RETURN
2930 '*STXY
2940      B(L)=X   : B(L+1)=Y   : L=L+2 : RETURN
2950 '*STLPXY
2960      B(L)=LPX : B(L+1)=LPY : L=L+2 : RETURN
2970 '*STDXY
2980      B(L)=DX  : B(L+1)=DY  : L=L+2 : RETURN
2990 '*STPCBC
3000      B(L)=PC  : B(L+1)=NBC : L=L+2
3010      FOR II=0 TO NBC: B(L)=BC(II): L=L+1: NEXT
 II: RETURN
3020 '*STMXY  ;(STORE SYMBOL MUL FACTOR)
3030      B(L)=MX  : B(L+1)=MY  : L=L+2 : RETURN
3040 '*GETXY
3050      X=B(L)   : Y=B(L+1)   : L=L+2 : RETURN
3060 '*GETLPXY
3070      LPX=B(L) : LPY=B(L+1) : L=L+2 : RETURN
3080 '*GETDXY
3090      DX=B(L)  : DY=B(L+1)  : L=L+2 : RETURN
3100 '*GETPCBC
3110      PC=B(L)  : NBC=B(L+1) : L=L+2
3120      FOR II=0 TO NBC: BC(II)=B(L): L=L+1: NEXT
 II: RETURN
3130 '*GETMXY
3140      MX=B(L)  : MY=B(L+1)  : L=L+2 : RETURN
3150 '*PUTGPD
3160      GOSUB 3040 : GOSUB 3060: GPNO=B(L) : L=L+
1 :'*GETXY,*GETLPXY
3170      LX=X+DXGP(GPNO): LY=Y+DYGP(GPNO)
3180      IF GPNO=0 THEN PUT@A(X,Y)-(LX,LY),GP0,PSE
T : GOTO 2200 :'*LPSET
3190      IF GPNO=1 THEN PUT@A(X,Y)-(LX,LY),GP1,PSE
T : GOTO 2200
3200      IF GPNO=2 THEN PUT@A(X,Y)-(LX,LY),GP2,PSE
T : GOTO 2200
```

リスト3　JUG-15プログラム・リスト

```
3210    IF GPNO=3 THEN PUT@A(X,Y)-(LX,LY),GP3,PSE
T : GOTO 2200
3220    IF GPNO=4 THEN PUT@A(X,Y)-(LX,LY),GP4,PSE
T : GOTO 2200
3230    IF GPNO=5 THEN PUT@A(X,Y)-(LX,LY),GP5,PSE
T : GOTO 2200
3240    IF GPNO=6 THEN PUT@A(X,Y)-(LX,LY),GP6,PSE
T : GOTO 2200
3250    IF GPNO=7 THEN PUT@A(X,Y)-(LX,LY),GP7,PSE
T : GOTO 2200
3260    IF GPNO=8 THEN PUT@A(X,Y)-(LX,LY),GP8,PSE
T : GOTO 2200
3270    IF GPNO=9 THEN PUT@A(X,Y)-(LX,LY),GP9,PSE
T : GOTO 2200
3280 '*RDGP
3290    ON ERROR GOTO 4460
3300    OPEN "I",#2,GPN$
3310    FOR JJ=0 TO 9:INPUT #3,DXGP(JJ): NEXT JJ
3320    FOR JJ=0 TO 9:        INPUT #3,DYGP(JJ): NEX
T JJ
3330    FOR JJ=0 TO GLIM : INPUT #3,GP0(JJ) : NEX
T JJ
3340    FOR JJ=0 TO GLIM : INPUT #3,GP1(JJ) : NEX
T JJ
3350    FOR JJ=0 TO GLIM : INPUT #3,GP2(JJ) : NEX
T JJ
3360    FOR JJ=0 TO GLIM : INPUT #3,GP3(JJ) : NEX
T JJ
3370    FOR JJ=0 TO GLIM : INPUT #3,GP4(JJ) : NEX
T JJ
3380    FOR JJ=0 TO GLIM : INPUT #3,GP5(JJ) : NEX
T JJ
3390    FOR JJ=0 TO GLIM : INPUT #3,GP6(JJ) : NEX
T JJ
3400    FOR JJ=0 TO GLIM : INPUT #3,GP7(JJ) : NEX
T JJ
3410    FOR JJ=0 TO GLIM : INPUT #3,GP8(JJ) : NEX
T JJ
3420    FOR JJ=0 TO GLIM : INPUT #3,GP9(JJ) : NEX
T JJ
3430    ON ERROR GOTO 0
3440    CLOSE #2 : RETURN
3450    '
3460    '*KANJID
3470      KC=B(L) : L=L+1:GOSUB 3140: GOSUB 3040
: '*GETMXY,*GETXY
3480      SYMBOL@ (X,Y),FNKAN$(KC), MX, MY, C
3490      X=X+16*MX : GOSUB 2200 : IF B(L)<>(-1)
THEN 3460 :'*LPSET
3500          L=L+1 : RETURN
3510      '*DUM
3520          RETURN
3530 '*Error
3540          RESUME 1310
3550 '*****************************
3560 '*    AA$ ｦ ﾊｯｾｲ(*HASSEI)        *
3570 '*****************************
3575 RETURN
3580     OPEN "I", #1,"COM0:(F8E1)"
3590     OPEN "O", #2,"COM0:(F8E1)"
3600     II=1:' POINTER IN AA$
3610     BB$=MID$(AA$,II,4)
3620     IF BB$="" THEN CLOSE #1,#2: RETURN
3630     GOSUB 3670: '*HASB
3640     GOTO 3610
3650  '
3660  ' HASB (BB$ ｦ ﾊｯｾｲ & POINTER ｺｳｼﾝ)
3670  '
3680     II=II+1
3690     B1$=MID$(BB$,1,1):IF B1$=" " OR B1$="｡"
OR B1$="､" THEN GOTO 4400:'*SLEE
P
3700     B2$=MID$(BB$,2,1):IF B1$="ﾊ" AND ( B2$="
 " OR B2$="､") THEN B1$="ﾜ"
3710     B3$=MID$(BB$,3,1)
3720     B4$=MID$(BB$,4,1)
3730     IF B2$="ﾞ" THEN 3880
3740     IF B2$="ﾟ" THEN 3880
3750     IF B2$="ｬ" THEN 3830
3760     IF B2$="ｭ" THEN 3830
3770     IF B2$="ｮ" THEN 3830
3780     IF B2$="-" THEN B1$=B1$+"-": GOSUB 4290:
 II=II+1: RETURN :'*SENDB1$
3790     IF B2$="ｯ" THEN B1$=B1$+"ｯ": GOSUB 4290:
 II=II+1: RETURN :'*SENDB1$
3800     B1$=B1$+CHR$(&HFF): GOSUB 4290: '*SENDB1
$
3810      RETURN
3820  '
3830  ' P ｬ ｭ ｮ
3840      II=II+1
3850      GOSUB 4010: 'CODE CONV. I
3860      B2$=B3$: GOTO 3780 :'P2 ﾍ
3870  '
3880 'P ﾞ ﾟ
3890      II=II+1
3900      IF B3$="ｬ" THEN 3950
3910      IF B3$="ｭ" THEN 3950
3920      IF B3$="ｮ" THEN 3950
3930      GOSUB 4130: 'CODE CONV. II
3940      B2$=B3$: GOTO 3780
3950      II=II+1
3960      GOSUB 4180: 'CODE CONV. III
3970      B2$=B4$: GOTO 3780
3980  '
3990  ' CODE CONV. I (ｷｬ, ｷｭ, ｷｮ, ...)
4000  '
4010      IF B1$="ｷ" THEN B1$=CHR$(&H80): GOTO 408
0
4020      IF B1$="ｼ" THEN B1$=CHR$(&H83): GOTO 408
0
4030      IF B1$="ﾁ" THEN B1$=CHR$(&H86): GOTO 408
0
4040      IF B1$="ﾆ" THEN B1$=CHR$(&H89): GOTO 408
0
4050      IF B1$="ﾋ" THEN B1$=CHR$(&H8C): GOTO 408
0
4060      IF B1$="ﾐ" THEN B1$=CHR$(&H8F): GOTO 408
0
4070      IF B1$="ﾘ" THEN B1$=CHR$(&H92): GOTO 408
0
4080      IF B1$="ｬ" THEN RETURN
4090      B1$=CHR$(ASC(B1$)+1)
4100      IF B2$="ｭ" THEN RETURN
4110      B1$=CHR$(ASC(B1$)+1): RETURN
4120  '
4130  ' CODE CONV. II (ｷﾞ, ﾊﾟ, ...)
4140  '
4150      IF B2$="ﾟ" THEN B1$=CHR$(ASC(B1$)+&H2B):
 RETURN
4160      B1$=CHR$(ASC(B1$)+&H30): RETURN
4170  '
4180  ' CODE CONV. IV (ｷﾞｬ, ﾋｬ, ...)
4190  '
4200      IF B1$="ｷ" THEN B1$=CHR$(&H95): GOTO 425
0
4210      IF B1$="ｼ" THEN B1$=CHR$(&H98): GOTO 425
0
4220      B1$=CHR$(&H9B)
4230      IF B2$="ﾞ" THEN 4250
4240      B1$=CHR$(ASC(B1$)+3)
4250      IF B3$="ｭ" THEN RETURN
4260      B1$=CHR$(ASC(B1$)+1)
4270      B1$=CHR$(ASC(B1$)+1): RETURN
4280  '
4290  ' SEND B1$ TO SIO-PORT 0
4300  '
4310      PRINT #2, B1$;
4320      COM(0) ON :' SIO PORT INTERUPT ENABLE
4330      GOTO 4330 :' WAIT FOR SIO-INTERUPT
4340      COM (0) OFF: RETURN
4350  ' ENTER HERE ON SIO-INPUT INTERUPT
4360      RETURN 4340
4370 '***********
4380 '*  SLEEP  *
4390 '***********
4400      FOR I1=0 TO 70: NEXT I1: RETURN
4410 '*SLEEP2
4420      FOR I1=0 TO 200: NEXT I1: RETURN
4430 '*******************
4440 '*  ERROR ｼｮﾘ ﾙｰﾁﾝ  *
4450 '*******************
4460 ' "SGP" NOT FOUND ERROR
4470      RESUME 3430
4480 ' "sindo" NOT FOUND ERROR
4490      RESUME 380
4500 '
4510      CONSOLE 0,24,1:COLOR 4:END
4520 ' ｵﾝｾｲ ｺﾞｳｾｲ ｿｳﾁｦ ｼﾖｳ ｽﾙﾙﾄｷﾊ 3575 ｦ ｹｽｺﾄ｡
4530 ' HARDC ｦ ﾄﾙﾄｷﾊ 1102, 965 , 1015, 519, 1415
 ﾉ "'" ｦ ｹｽｺﾄ｡
4540 ' Light Pen ｦ ｼﾖｳ ｽﾙﾄｷﾊ 630, 1560 ｦ REM ﾌﾞﾝ
ﾆ ｶｴﾙｺﾄ｡
4550 ' SAVE "JUG-15
```

■六田嘉明

本プログラムは，画面に図形・文字（漢字を含む）を描き，これをデジタル・プロッタMIPLOT-II（MP1000-21型，グラフテック社）に送り出すプログラムです．FM-11のミニ・フロッピーディスクおよび640×400ドットモードのディスプレイを想定しています(CCOPYまたはWCCOPYとシャープのカラープリンタを使うと，カラーハード・コピーもとることができます．

本プログラムは，2つのモードに分かれています．PENは画面作成用，PLOTTERは，MIPLOT-II出力用です．

表１に示した31個のコマンドを使って，画面に図形や文字を描きます．作成した画面は，ファイルにセーブしておき，PEN，PLOTTER，および，別稿のCAI (p.4)プログラムで使います．

使用法は，表１を見ればわかると思いますが，2，3の注意事項を次に述べます．

❶コマンド〝5″(数字キーの5)，〝.″

Last Point(LP)をその点に設定するコマンドです．ただし，[5]では画面に点を打たず，[.]で画面に点を打ちます．

❷コマンド〝+″

カーソルを水平，垂直方向に延長します．位置を確かめるために用います．XORを使っているので，線上に点や線があると，その部分が消えます．これで，正確にカーソルが一致したかどうかがわかります．1～2秒でこのカーソル延長線は消え，元の画面に戻ります．

❸コマンド〝d″，〝0″，〝1″

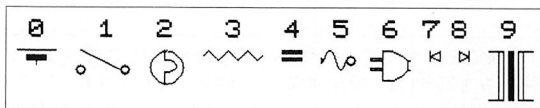
使用頻度の高い図形をファイルに登録するには，先ず，その図形の右下端にカーソルを合わせ，[5]を押します．次に，カーソルを図形の左上端に移動し，[d]を入力するとパターン番号を問い合わせてくるので，0～9の数を入力します．以上で，図形番号の定義ができました．[／]を押すと，グラフィック・パターン用ファイルの名前を聞いてくるので，これに答えるとその名前で，0～9までのパターンがファイルに登録されます．

グラフィック・パターンを画面にだしたいときは，カーソルをパターンの左上端にもってきて，[o]を押します．つぎに，パターン番号を記入すると，そのパターンがカーソル位置を左上端にして表示されます．

表１　画面編集プログラムのコマンド

（＊印は，プロッタ出力時に無効となるコマンド）

分類	＊	キー	機能
カーソル関係		7 8 9 / 4 ← 5 → 6 / 1 2 3	カーソル(c)を動かす
		＋	カーソルを縦，横に延長．
		s	cの移動量をドット単位で指定．
グラフィックス関係		5	Last Pointをセットする．
		・	点を打つ．
	＊	0	点を消す．
		a	円弧を描く．
		b	箱を描く．
		c	円を描く．
	＊	d	グラフィック図形を定義する．
	＊	o	グラフィック図形を描く．
	＊	／	グラフィック図形をファイルに格納．
	＊	g	グラフィック図形をGETする．
	＊	m	GETしたグラフィック図形をPUTする．
		p	色を塗る．
		SPACE	色(文字および線)を変える．
文字	＊	k	漢字入力モードに入る．
		r	英，数，カナ，記号入力モードに入る．
その他		w	全入力情報をファイルへ格納．
		x	入力を取り消す．
		コントロールA	終了する．

図１　登録できるグラフィックパターンの例

❹コマンド〝g″，〝m″

図形をgコマンドで配列へgetします．mコマンドは任意の位置にgetした図形をputすることができます．カーソルの合わせ方は，d，oの場合と同じです．

❺コマンド〝a″

円弧を描きます．角度は中心の真右の円周上の点を0°とし，反時計方向に0°から360°まで指定します．たとえば，90，180と指定すると，第２象限内の1/4円弧が，180，90と折定しると，180°から360°＝0°を通って90°までの3/4円弧が描かれます．

❻コマンド〝K″

漢字入力モードになります．音読みの頭１字をカナで入力し，表示された漢字の中から相当するものを番号で指定する，という原始的な方法をとっています．ただし，FM-

図2　作図画面のハードコピー

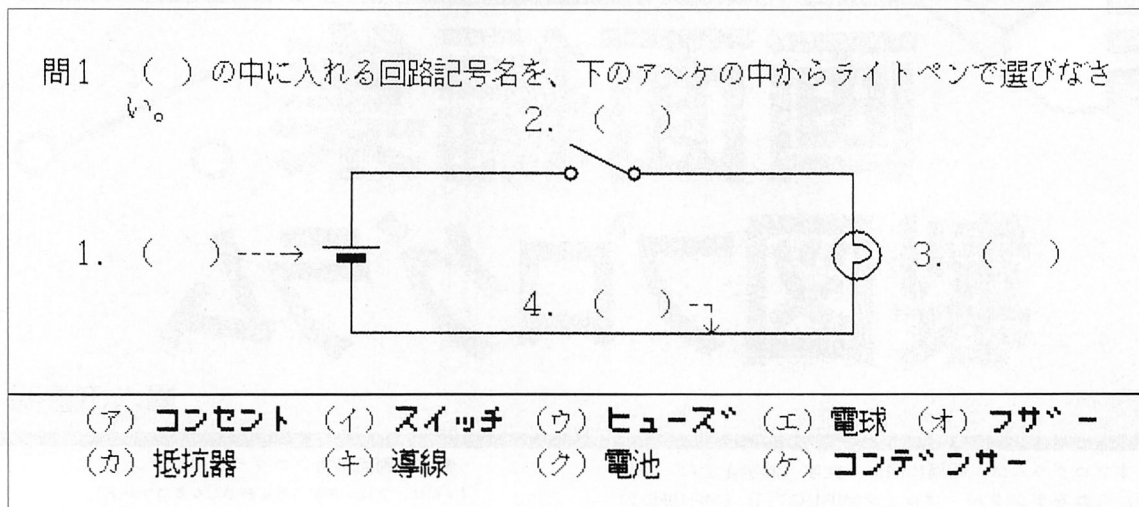
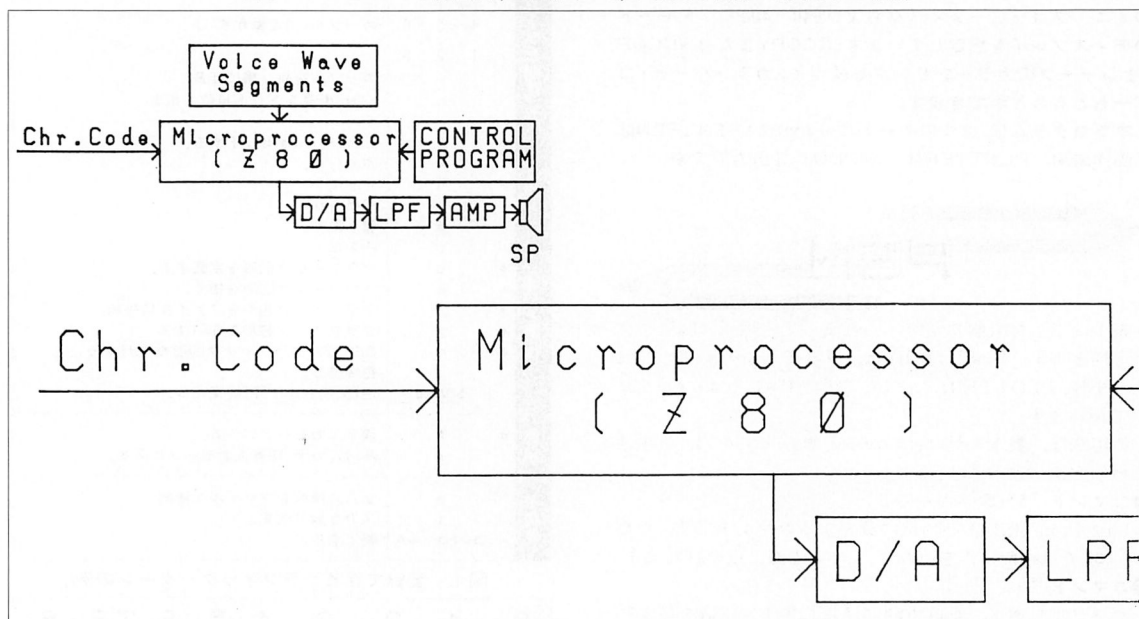
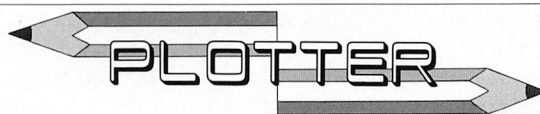



図3　MIPLOTT-II作図例（上は倍率＝2，下は同じ図形の倍率6.5の場合）

11の機能を生かして，色や縦，横の倍率を指定できます（⏎のみを返せば，以前に指定した値がとられます）．なお，S0からS49までを入力すると，その番号で登録された漢字（1420〜1470行）が取りだされます．

❼コマンド "x"

訂正用コマンドです．本プログラムは，入力したコマンドを入力した順に配列に格納していくので，誤まって入力したときは，x を押し，続いて何個バックするかを入力します．このとき，画面は一担消去され，初めから描きなおします．

また，初めの方で入力したものを消去したいときは，そのステップまで戻すことになり，大変なロスです．もっと，改良しなければならないと考えています．

終了は，CTRL＋A を入力します．Break キーで終了させると，画面情報がファイルにセーブされないので，くれぐれも注意してください．

PENで作成した画面をファイルから読み出し，プロッタに出力するプログラムです．MIPLOTT-II（グラフテック社）は，FM-11のプリンタ・ポートに，そのまま接続できます．

プロッタでは，漢字が描けませんし，配列にセーブされたパターンは，ドット情報として格納されているので，これを描くことはできません．したがって，これらのコマンドは読み飛ばされます．読み飛ばされる（無効扱い）コマンドは，**表1**に＊印で示してあります．

なお，ディスプレイ上ではカラーは7色であるのに，MIPLOT-IIではペンが6本ですから，COLOR1と5は，5番ペンで，COLOR7（白）は第1番ペンで描くことにし，その他はCOLOR番号とペン番号が同じにしてあります．ただ

し，黒色モードでは，すべての色が1番ペンで描かれます．

MIPLOT-IIは，円，円弧，カナを描くことができないので，これらはソフトで描かせることにしました．ただし，カナをかかせるにはカナ・フォント(字体)を**DATA**文で定義しておく必要があります．今回のプログラムでは，『ア』と『イ』のフォントのみが6620，6630行にあります．フォント・データの作製は，この例を参照して行なってください．

この行にある**R**は**Relative Mode**命令で，現在のペン位置からx方向，y方向への移動量をドット単位で与えます．

そして**I**は**Relative Draw**命令で，文法はI, Δx_1, Δy_1, Δx_2, Δy_2, ……の形です．このとき，現在位置(x_0, y_0)から$(x_0+\Delta x_1, y_0+\Delta y_1)\rightarrow(x_0+\Delta x_1+\Delta x_2, y_0+\Delta y_1+\Delta y_2)\rightarrow\cdots$と順に線分が引かれます．コマンドの終わりは，Eでで示します．

最初のペン位置は，文字の左下端にあるものとし，1文字を16×16ドットの中の横12×縦14ドットのわく内で表わし，データ文を作製します．最後に，必ずペンを次の文字の左下端へ移動する命令を入れておいてください．

プロット時の倍率は，1から6.5倍まで指定できます．4倍で，スクリーン1画面分の図形がA4用紙1ページに入ります．私はもっぱら，簡易自動製図機として，論文などの図面作成用に活用しています．MIPLOT-IIがない場合，ドット・プリンタ(エプソンMP-80，82など)などハードコピーでも，ある程度の役には立ちますし，この場合には漢字でレタリングした図面が作れることが，メリットと言えましょう．

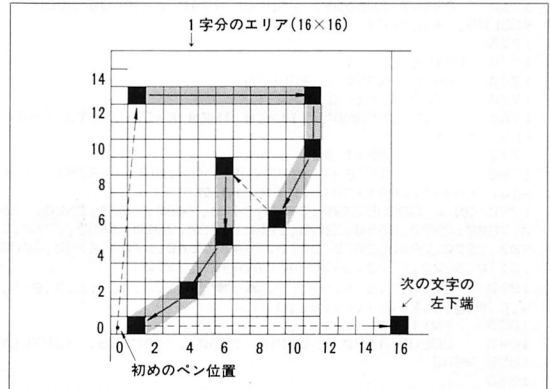



図4　カナ・フォント作製例

(DATA "R, 1, 13, I, 10, 0, 0, −3, −2, −4, R, −3, 3, I, 0, −4, −2, −3, −3, −2, R, 15, 0, E")

画面編集・製図プログラムPen Plt-2

```
1000 PRINT TAB(15);"****************************
****************
1010 PRINT TAB(15);"*    SCREEN EDITOR AND PLOTTE
R --PENPLOT--   *
1020 PRINT TAB(15);"*         FOR FM-11 AND MIPLO
T-II            *
1030 PRINT TAB(15);"*        CODED BY Y. MUDA (19
83.9.19)        *
1040 PRINT TAB(15);"****************************
****************
1050 PRINT TAB(15);"             JUST A MOMENT, PLE
ASE!"
1060 DEFINT A-Z:DEFSNG T,R,M
1070  KLIM=50:KCLIM=59:YMAX=2700:CMN=33:ALIM=400
:BLIM=100:CLIM=25:DLIM=1700:GLIM=300
1080 DIM CU%(11),CURSV%(200),CURSH%(121),PICTURE
%(GLIM),CMMAND%(CMN)
1090 DIM GP0(GLIM),GP1(GLIM),GP2(GLIM),GP3(GLIM)
,GP4(GLIM)
1100 DIM GP5(GLIM),GP6(GLIM),GP7(GLIM),GP8(GLIM)
,GP9(GLIM)
1110 DIM KCODE%(KLIM),KCD(KCLIM),A(ALIM),TA(BLIM
),A$(CLIM),B(DLIM)
1120          A,TA,A$,B=storange for input data.
I,J,K,L=pointer for A,TA,A$,B
1130 DEF FNKAN$(KC)= CHR$(KC ¥ &H100) +CHR$(KC M
OD &H100)
1140 I=0:J=0:K=0:L=0:C=7: GXYMAX=GLIM/3*16: MX=1
:MY=1
1150 MDRF=3.14159/180: 'Conversion Factor From D
egree To Radian
1160 INC0=16:INC1=16:X=19:Y=48:GOSUB 2750:GOSUB
2750:'*LPSET,*LPSET
1170
1180 RESTORE 1200
1190 FOR II=1 TO CMN : READ CMMAND%(II) : NEXT
1200 'COMDATA
1210 DATA &H20,&H73,&H30,&H6c,&H62,&H63,&H1c,&H1
d,&H1e,&H1f
1220 '       ,    s,    0,    1,    b,    c,   ->,  <
-,   ▲,   ▼
1230 DATA &H31,&H32,&H33,&H34,&H35,&H36,&H37,&H3
8,&H39,&H77
1240 '        1    2    3    4    5    6    7
8    9    w
1250 DATA &H70,&H6D,&H67,&H78,&H6B,&H61,&H2B,&H7
2,&H69,&H2E
1260 '        p    m    g    x    k    a    +
r    i    .
1270 DATA &H64,&H6f,&H2f
1280 '        d    o    /
1290 '
1300 RESTORE 1330:'KCDEDATA
1310 FOR II=0 TO KLIM: READ KCODE%(II) : NEXT II
1320 'KCODEDATA
1330 DATA &H0121,&H2121,&H2330,&H2421,&H2521,&H2
621,&H2721
1340 DATA &H3021,&H304a,&H3126,&H3143,&H3177,&H3
23c,&H346b,&H3665,&H3735,&H3843
1350 DATA &H3a33,&H3b45,&H3f5a,&H4024,&H4139,&H4
23e,&H434d,&H4445,&H4462,&H4546
1360 DATA &H4660,&H4673,&H4728,&H4729,&H4735,&H4
743,&H485d,&H4954,&H4a3a,&H4a5d
1370 DATA &H4b60,&H4c23,&H4c33,&H4c3d,&H4c4e,&H4
c69,&H4c7b,&H4d3d
1380 DATA &H4d65,&H4d78,&H4e5c,&H4e61,&H4f24,&H4
f41
1390 RESTORE 1420:'KCDATA
1400 FOR II=0 TO KCLIM : READ KCD(II) : NEXT II
1410 '*KCDATA (S0 TO S59)
1420 DATA &H3020,&H3035,&H304C,&H3221,&H3230,&H3
273,&H3259,&H3456,&H3425,&H3524
1430 DATA &H346F,&H3565,&H3757,&H376B,&H383B,&H3
933,&H3872,&H3E48,&H3E43,&H3F5E
1440 DATA &H407E,&H422C,&H424E,&H4331,&H434D,&H4
353,&H4434,&H443E,&H444C,&H4545
1450 DATA &H4471,&H446A,&H4626,&H4630,&H4633,&H4
574,&H475B,&H482F,&H4969,&H4A42
1460 DATA &H4C23,&H4C40,&H4D36,&H4D51,&H4D46,&H4
D6E,&H4E2E,&H4E4F,&H4E2C,&H214D
1470 DATA &H4E4C,&H4E28,&H4E73,&H4E63,&H4F29,&H4
F3F,&H214A,&H214B,&H214C,&H214D
1480
1490      PRINT: INPUT "PEN (1), PLOTTER(2) ";PP
1500      ON PP GOTO 1540, 5700 : '*PEN, PLOTTER
1510       '****************
1520       '*    P E N     *
1530       '****************
1540 OPEN "I",#1,"KYBD:"
1550 ON ERROR GOTO 5620:'*ERRO
1560 WIDTH 80,25: COLOR 7,0,7,0 :CLS
1570 LINE (0,3)-(6,3), PSET: LINE (3,0)-(3,6),PS
ET:GET@A (0,0)-(6,6),CU%,G
1580 PUT@A (0,0)-(6,6),CU%,XOR :'GET GCURSOR
1590 '
1600 CONSOLE 0,2,0
1610 LINE (3,3)-(636,396),PSET,7,B
1620 GET@A (3,3)-(3,396),CURSV%,G:'get vertical
cursur
1630 GET@A (3,3)-(636,3),CURSH%,G:'get horizonta
l cursor
1640  LINE (3,3)-(636,396),PRESET,7,B
1645   INPUT "ﾜｸｦ ｶｷﾏｽｶ (y/n)";FRM$:IF FRM$="y"
THEN GOSUB 2065:'*FRAME
1650   INPUT "File Name (New File=n) ";DTNM$
1660   INPUT "Circle Factor(0-1, 0=default[0.899
0])";MFCIRC!
1670    IF MFCIRC!=0 THEN MFCIRC!=.899
1680   INPUT "Name of your GP (n =no GP)";GPN$
1690   IF GPN$ <>"n" AND GPN$ <>"N" THEN GOSUB 4
590 :'*RDGP
```

```
1700   IF DTNM$<>"n" AND DTNM$<>"N" THEN GOSUB 3
420 :'*DRAWF
1710   PRINT "READY" :GOSUB 1990 : GOSUB 2000 :'
*GCURS, *GETKEY
1720 '
1730  WHILE CH<>1
1740   GOSUB 1990 :'*GCURS
1750    FOR II=1 TO CMN
1760     IF CMMAND%(II)=CH THEN F=II :GOTO 1780
ELSE F=-1
1770         NEXT II
1780         IF F=-1 THEN I=I-1 :BEEP:PRINT "Ill
egal com-";CHR$(CH) :GOTO 1830
1790 ON F GOSUB2390,2960,2440,2460,2480,2500,276
0,2800,2820,2850,2890,2850,2920,2800,3650,2760,2
900,2820,2940,2070,2530,2640,2660,2700,4790,3000
,3130,3200,3280,2420,4270,4420,2210
1800 ' ,s,0,l,b,c,->,<-,▲,▼,1,2,3,4,5,6,7,8,9,
w,p,m,g,x,k,a,+,r,i,.,d,o,/
1830   '*NXTCOM
1840   GOSUB 1990 : GOSUB 2000 :'*GCURS, *GETKEY
1850 WEND
1860 '
1870  GOSUB 1990 :'*GCURS-OFF
1880  INPUT "Screen / Hard Copy ｦ ﾄﾘﾏｽｶ (Y/N)";H
C$:IF HC$="y" THEN HC$="Y"
1890  IF HC$<>"Y" THEN 1940
1900   INPUT "MODE(0=text,1=g.ｶｸﾀﾞｲ,2=g.ｼｭｸ,3=t+
g.ｶｸ,4=t+g.ｼｭｸ)";HCM:IF HCM<0 OR HCM>4 THEN 1900
1910    PRINT "Press any key." :POKE &HFC12,&H17
 :'400 dots mode HCOPY ﾖｳ
1920     IF INKEY$="" THEN 1920 ELSE HARDC HCM
1930 '
1940      '*Bye
1950      PRINT"See you again !"
1960      I=I+1: GOSUB 2070 :CONSOLE 0,24,1: COL
OR 4: '*WR
1970    END
1980 '*GCURS
1990    PUT@A(X-3,Y-3)-(X+ 3,Y+ 3),CU%,XOR:RETUR
N
2000 '*GET KEY
2010    CH$=INKEY$: IF CH$="" GOTO 2010 ELSE CH=
ASC(CH$)
2020    A(I)=CH : I=I+1
2030    IF I= (ALIM-10) OR J>(BLIM-10) OR K>(CLI
M-1) OR L>(DLIM-15) THEN GOSUB 2050 : '*CAUTION
2040          RETURN
2050 '*CAUTION
2060   PRINT "STORANGE-FULL !  type CNTL-A for s
ave":BEEP:RETURN
2064 '*FRAME
2065   LINE(3,32)-(637,397),PSET,7,B:LINE(3,288)
-(637,288),PSET,7
2067   RETURN
2070 '*WR
2080   I=I-1
2090   INPUT "File Name (0=not save)";GDAT$:IF G
DAT$="0" THEN RETURN
2100   IF GDAT$=DTNM$ THEN KILL DTNM$
2110   OPEN "O",#3,GDAT$
2120   JJ=J : IF J<>0 THEN JJ=J-1
2130   KK=K : IF K<>0 THEN KK=K-1
2140   LL=L : IF L<>0 THEN LL=L-1
2150   PRINT #3,I-1,JJ,KK,LL
2160   FOR II=0 TO I-1 : PRINT #3,A(II) : NEXT
2170   FOR II=0 TO JJ  : PRINT #3,TA(II): NEXT
2180   FOR II=0 TO KK  : PRINT #3,A$(II): NEXT
2190   FOR II=0 TO LL  : PRINT #3,B(II) : NEXT
2200   CLOSE #3 : RETURN
2210 '*Write GP
2220   INPUT "GP File Name";GPN1$
2230   IF GPN1$="" OR GPN1$=GPN$ THEN KILL GPN$:
 GPN1$=GPN$
2240   OPEN "O",#3,GPN1$
2250   FOR JJ=0 TO 9 : ! #3,DXGP(JJ):NEXT JJ
2260   FOR JJ=0 TO 9 : ! #3,DYGP(JJ):NEXT JJ
2270   FOR JJ=0 TO GLIM : ! #3,GP0(JJ) : NEXT JJ
2280   FOR JJ=0 TO GLIM : ! #3,GP1(JJ) : NEXT JJ
2290   FOR JJ=0 TO GLIM : ! #3,GP2(JJ) : NEXT JJ
2300   FOR JJ=0 TO GLIM : ! #3,GP3(JJ) : NEXT JJ
2310   FOR JJ=0 TO GLIM : ! #3,GP4(JJ) : NEXT JJ
2320   FOR JJ=0 TO GLIM : ! #3,GP5(JJ) : NEXT JJ
2330   FOR JJ=0 TO GLIM : ! #3,GP6(JJ) : NEXT JJ
2340   FOR JJ=0 TO GLIM : ! #3,GP7(JJ) : NEXT JJ
2350   FOR JJ=0 TO GLIM : ! #3,GP8(JJ) : NEXT JJ
2360   FOR JJ=0 TO GLIM : ! #3,GP9(JJ) : NEXT JJ
2370   CLOSE #3 : RETURN
2380 '*Color
2390   INPUT"Color (0-7) ";C:IF C<0 OR C>7 THEN
BEEP : GOTO 2390
2400   B(L)=C : L=L+1 : COLOR C,0,C,0 : RETURN
2410 '*PSETR
2420   GOSUB 4040 : PSET (X,Y,C) : GOTO 2750 :'*
STXY,*LPSET
2430 '*PRESETR
2440   GOSUB 4040 : PRESET(X,Y,C): GOTO 2750
2450 '*LINER
2460   GOSUB 4040 : GOSUB 4060 : GOTO 3700
2470 '*BOXR
2480   GOSUB 4040 : GOSUB 4060 : GOTO 3740
2490 '*CIRCLER
2500   RADIUS=SQR((X-LPX)^2+(Y-LPY)^2):IF RADIUS
=0 THEN 4740:'*ERR-B
2510   TA(J)=RADIUS : TA(J+1)=MFCIRC!: J=J+2 : T
S=0:TE=1:GOSUB 4040: GOTO 3810:'*STXY,*CIRCLEJ
2520 '*PAINTR
2530   INPUT "Paint color (0-7)";PC: IF PC<0 OR
PC>7 THEN 4740:'*ERR-B
2540   INPUT "Border colors(1-7)[ex. 7 or 1/2/3
etc.]";BC$
2550   PMID=1: BCEND=0
2560    FOR II=0 TO 6: BCP$=""
2570     BCPM$=MID$(BC$,PMID,1): IF BCPM$="" THE
N BCEND=1: GOTO 2590
2580      IF BCPM$<>"/" THEN BCP$=BCP$+BCPM$ :PM
ID=PMID+1: GOTO 2570
2590       BC(II)=VAL(BCP$): PMID=PMID+1:IF BCEN
D=1 GOTO 2610
2600      NEXT II
2610    NBC=II
2620   GOSUB 4100 : GOSUB 4040 : GOTO 3850 :'*S
TPCBC,*STXY,*PAINTJ
2630 '*PUTR ;PUT PICTURE%
2640   GOSUB 4040 : GOSUB 4090 : GOTO 3910 :'*S
TXY,*STDXY,*PUTJ
2650 '*GETR   ;GET PICTURE%
2660   IF ABS((X-LPX)*(Y-LPY))>GXYMAX THEN 4740
: '*ERR-B
2670   DX=LPX-X: DY=LPY-Y
2680   GOSUB 4040 : GOSUB 4090 : GOTO 3950 :'*S
TXY,*STDXY,*GETRJ
2690 '*CANR
2700   PRINT "CANCEL-How many commands(0-";I-1;
")";:INPUT II
2710   IF II<0 OR II>I-1 THEN GOTO 4740 :'*ERR-
B
2720   A(I-II-1)=1
2730   I=0:J=0:K=0:L=0:CLS 0: GOTO 3440:'*DRAW
2740 '*LPSETR
2750   LLPX=LPX : LLPY=LPY : LPX=X : LPY=Y : RE
TURN
2760 '*CURR
2770   X=X+INC0 : IF X>636 THEN X=3
2780   GOTO 4760
2790 '*CURL
2800   X=X-INC0 : IF X<3 THEN X=636
2810   GOTO 4760
2820 '*CURU
2830   Y=Y-INC1 : IF Y<3 THEN Y=396
2840   GOTO 4760
2850 '*CURD
2860   Y=Y+INC1 : IF Y>396 THEN Y=3
2870   GOTO 4760
2880 '*CURLD
2890   I=I+1 : GOSUB 2800 : GOTO 2850
2900 '*CURLU
2910   I=I+1 : GOSUB 2800 : GOTO 2820
2920 '*CURRD
2930   I=I+1 : GOSUB 2760 : GOTO 2850
2940 '*CURRU
2950   I=I+1 : GOSUB 2760 : GOTO 2820
2960 '*STEPR
2970   INPUT "Step (1-256)";INC0: INC1=INC0
2980   IF INC0<0 OR INC0>256 THEN BEEP : GOTO 29
60
2990   GOTO 4760 :'*NOSAVE
3000 '*ARCR
3010   INPUT "ｶｲｼ ｶｸﾄﾞ, ｼｭｳﾘｮｳ ｶｸﾄﾞ(0-360)";TE,T
S
3020   TS=1-TS/360: TE=1-TE/360
3030   RADIUS=SQR((X-LPX)^2+(Y-LPY)^2):IF RADIUS
=0 THEN 4740:'*ERR-B
3040   TA(J)=RADIUS : TA(J+1)=MFCIRC!
3050   TA(J+2)=TS: TA(J+3)=TE: J=J+4: GOSUB 4040
: '*STXY
3060   GOTO 3810: '*CIRCLEJ
3070 '*CONV
```

```
3080    TH=TH/(2*3.14159265#)
3090    IF XX>=XC AND YY>=YC THEN TH=1-TH   : RET
URN:' I ｼｮｳｹﾞﾝ
3100    IF XX>=XC AND YY< YC THEN TH=-(1+TH): RET
URN:' IV
3110                TH=-(.5+TH): RETURN
           :' II, III
3120 '*CURSR
3130    PUT@A (X,3)-(X,396),CURSV%,XOR : PUT@A (3
,Y)-(636,Y),CURSH%,XOR
3140    LINE (3,3)-(636,396),XOR,7,B
3150    FOR II=0 TO 4000:NEXT
3160    PUT@A (X,3)-(X,396),CURSV%,XOR : PUT@A (3
,Y)-(636,Y),CURSH%,XOR
3170    LINE (3,3)-(636,396),XOR,7,B
3180    GOTO 4760 :'*NOSAVE
3190 '*ALPH
3200    INPUT "Chars?";C$: IF C$="" THEN 4760 :'*
NOSAVE
3210    INPUT "ﾖｺ ﾊﾞｲﾘﾂ(1-40) or ret";MX$
3220     IF MX$="" THEN 3250 ELSE MX=VAL(MX$)
3230     INPUT "ﾀﾃ ﾊﾞｲﾘﾂ(1-25) or ret";MY$
3240     IF MY$="" THEN 3250 ELSE MY=VAL(MY$)
3250    A$(K)=C$ : K=K+1 :GOSUB 4140: GOSUB 4040
:'*STMXY,*STXY
3260    GOTO 4030 :'*ALPHJ
3270 '*INFO
3280    PRINT "    I     J     K     L     C     X
Y  LPX  LPY LLPX LLPY"
3290    PRINT USING " ####";I,J,K,L,C,X,Y,LPX,LPY
,LLPX,LLPY;
3300    I=I-1 : RETURN
3310 '
3320 '*RDFILE
3330    OPEN "I",#2,DTNM$
3340    INPUT #2,LIM,LIB,LIC,LID
3350    FOR II=0 TO LIM : INPUT #2,A(II)  : NEXT
II
3360    FOR II=0 TO LIB : INPUT #2,TA(II) : NEXT
II
3370    FOR II=0 TO LIC : INPUT #2,A$(II) : NEXT
II
3380    FOR II=0 TO LID : INPUT #2,B(II)  : NEXT
II
3390    RETURN
3400 '**********************
3410 '*    start DRAWING    *
3420 '**********************
3430    GOSUB 3330:'*RDFILE
3440    C=7 : COLOR 7,0,7
3450    GOSUB 3580 :  '*GetKeyD
3460       WHILE CH<>1
3470        FOR II=1 TO CMN
3480          IF CMMAND%(II)=CH THEN F=II :GOTO 35
00 ELSE F=-1
3490       NEXT II
3500 ON F GOSUB 3600,2960,3660,3680,3720,3780,27
60,2800,2820,2850,2890,2850,2920,2800,3650,2760,
2900,2820,2940,2070,3830,3890,3930,2700,5550,397
0,3130,4000,3280,3620,5370,4450,5600
3510 's,0,l,b,c,->,<-,▲,▼
3520 '1 2 3 4 5 6 7 8 9 w
3530 'p m g x k a + r i . d o /
3540  GOSUB 3580 :'*GETKEYD
3550  WEND
3560  I=I-1 : CLOSE #2 : PRINT "READY" : RETURN
3570 '
3580 '*GETKEYD
3590    CH=A(I) : I=I+1 : RETURN
3600 '*COLORD
3610    C=B(L) : L=L+1 : COLOR C,0,C : RETURN
3620 '*PSETD
3630    GOSUB 4150 : PSET (X,Y,C):'*GETXY
3640 '*PSETJ
3650    GOTO 2750 :'*LPSET
3660 '*PRESETD
3670    GOSUB 4150 : PRESET (X,Y,C): GOTO 2750 :'
*GETXY,*LPSET
3680 '*LINED
3690    GOSUB 4150 : GOSUB 4170 :'*GETXY,*GETLPXY
3700 '*LINEJ
3710    LINE(LPX,LPY)-(X,Y),PSET, C: GOTO 2750 :'
*LPSET
3720 '*BOXD
3730    GOSUB 4150 : GOSUB 4170 :'*GETXY,*GETLPXY
3740 '*BOXJ
3750    LINE(LPX,LPY)-(X,Y),PSET,C,B: GOTO 2750 :
'*LPSET
3760 '
3770 '*CIRCLED
3780    RADIUS=TA(J): MFCIRC!=TA(J+1): J=J+2: GOS
UB 4150 :'*GETXY
3790      TS=0: TE=0
3800 '*CIRCLEJ
3810    CIRCLE(X,Y),RADIUS,C,MFCIRC!,TS,TE : GOTO
 2750 :'*LPSET
3820 '*PAINTD
3830    GOSUB 4210: GOSUB 4150 :'*GETPCBC,*GETXY
3840 '*PAINTJ
3850    FOR II=NBC+1 TO 6: BC(II)=BC(NBC): NEXT I
I :PRESET (X,Y)
3860    PAINT (X,Y),PC,BC(0),BC(1),BC(2),BC(3),BC
(4),BC(5),BC(6)
3870    GOTO 2750: '*LPSET
3880 '*PUTD
3890    GOSUB 4150 :GOSUB 4200 :'*GETXY,*GETDXY
3900 '*PUTJ
3910    PUT@A (X,Y)-(X+DX,Y+DY),PICTURE%,PSET : G
OTO 2750 :'*LPSET
3920 '*GETD
3930    GOSUB 4150: GOSUB 4200 :'*GETXY,*GETDXY
3940 '*GETRJ
3950    GET@A (X,Y)-(X+DX,Y+DY),PICTURE%,G: GOTO
2750 :'*LPSET
3960 '*ARCD
3970    GOSUB 4160: '*GETXY
3980    RADIUS=TA(J) : MFCIRC=TA(J+1): TS=TA(J+2)
 : TE=TA(J+3) : J=J+4
3990    GOTO 3810: '*CIRCLEJ
4000 '*ALPHD
4010    C$=A$(K): K=K+1: GOSUB 4250: GOSUB 4150:
'*GETMXY,*GETXY
4020 '*ALPHJ
4030    SYMBOL (X,Y),C$,MX,MY,C : RETURN
4040 '*STXY
4050    B(L)=X   : B(L+1)=Y   : L=L+2 : RETURN
4060 '*STLPXY
4070    B(L)=LPX : B(L+1)=LPY : L=L+2 : RETURN
4080 '*STDXY
4090    B(L)=DX  : B(L+1)=DY  : L=L+2 : RETURN
4100 '*STPCBC
4110    B(L)=PC  : B(L+1)=NBC : L=L+2
4120    FOR II=0 TO NBC: B(L)=BC(II): L=L+1: NEXT
 II: RETURN
4130 '*STMXY  ;(STORE SYMBOL MUL FACTOR)
4140    B(L)=MX  : B(L+1)=MY  : L=L+2 : RETURN
4150 '*GETXY
4160    X=B(L)   : Y=B(L+1)   : L=L+2 : RETURN
4170 '*GETLPXY
4180    LPX=B(L) : LPY=B(L+1) : L=L+2 : RETURN
4190 '*GETDXY
4200    DX=B(L)  : DY=B(L+1)  : L=L+2 : RETURN
4210 '*GETPCBC
4220    PC=B(L)  : NBC=B(L+1) : L=L+2
4230    FOR II=0 TO NBC: BC(II)=B(L): L=L+1: NEXT
 II: RETURN
4240 '*GETMXY
4250    MX=B(L)  : MY=B(L+1)  : L=L+2 : RETURN
4260 '
4270 '*DEFGP
4280    IF ABS((X-LPX)*(Y-LPY)) > GXYMAX THEN 474
0: '*ERR-B
4290    INPUT "Pattern NO.(0-9)";GPNO
4300    IF GPNO=0 THEN GET@A(X,Y)-(LPX,LPY),GP0,G
 : GOTO 4410
4310    IF GPNO=1 THEN GET@A(X,Y)-(LPX,LPY),GP1,G
 : GOTO 4410
4320    IF GPNO=2 THEN GET@A(X,Y)-(LPX,LPY),GP2,G
 : GOTO 4410
4330    IF GPNO=3 THEN GET@A(X,Y)-(LPX,LPY),GP3,G
 : GOTO 4410
4340    IF GPNO=4 THEN GET@A(X,Y)-(LPX,LPY),GP4,G
 : GOTO 4410
4350    IF GPNO=5 THEN GET@A(X,Y)-(LPX,LPY),GP5,G
 : GOTO 4410
4360    IF GPNO=6 THEN GET@A(X,Y)-(LPX,LPY),GP6,G
 : GOTO 4410
4370    IF GPNO=7 THEN GET@A(X,Y)-(LPX,LPY),GP7,G
 : GOTO 4410
4380    IF GPNO=8 THEN GET@A(X,Y)-(LPX,LPY),GP8,G
 : GOTO 4410
4390    IF GPNO=9 THEN GET@A(X,Y)-(LPX,LPY),GP9,G
 : GOTO 4410
4400    GOTO 4760 :'*ERR-B
4410    DXGP(GPNO)=LPX-X: DYGP(GPNO)=LPY-Y: GOTO
4760: '*NOSAVE
4420 '*PUTGP
4430    INPUT "Pattern NO.(0-9)";GPNO : IF GPNO>9
```

```
 THEN I=I-1:BEEP:RETURN
4440   GOSUB 4040 : GOSUB 4060: B(L)=GPNO : L=L+
1 : GOTO 4470 :'*STXY, *STLPXY,*PUTGPJ
4450 '*PUTGPD
4460   GOSUB 4150 : GOSUB 4170: GPNO=B(L) : L=L+
1 :'*GETXY,*GETLPXY
4470 '*PUTGPJ
4480   LX=X+DXGP(GPNO): LY=Y+DYGP(GPNO)
4490   IF GPNO=0 THEN PUT@A(X,Y)-(LX,LY),GP0,PSE
T : GOTO 2750 :'*LPSET
4500   IF GPNO=1 THEN PUT@A(X,Y)-(LX,LY),GP1,PSE
T : GOTO 2750
4510   IF GPNO=2 THEN PUT@A(X,Y)-(LX,LY),GP2,PSE
T : GOTO 2750
4520   IF GPNO=3 THEN PUT@A(X,Y)-(LX,LY),GP3,PSE
T : GOTO 2750
4530   IF GPNO=4 THEN PUT@A(X,Y)-(LX,LY),GP4,PSE
T : GOTO 2750
4540   IF GPNO=5 THEN PUT@A(X,Y)-(LX,LY),GP5,PSE
T : GOTO 2750
4550   IF GPNO=6 THEN PUT@A(X,Y)-(LX,LY),GP6,PSE
T : GOTO 2750
4560   IF GPNO=7 THEN PUT@A(X,Y)-(LX,LY),GP7,PSE
T : GOTO 2750
4570   IF GPNO=8 THEN PUT@A(X,Y)-(LX,LY),GP8,PSE
T : GOTO 2750
4580   IF GPNO=9 THEN PUT@A(X,Y)-(LX,LY),GP9,PSE
T : GOTO 2750
4590 '*RDGP
4600   OPEN "I",#3,GPN$
4610    FOR JJ=0 TO 9:      INPUT #3,DXGP(JJ): NE
XT JJ
4620    FOR JJ=0 TO 9:      INPUT #3,DYGP(JJ): NE
XT JJ
4630    FOR JJ=0 TO GLIM : INPUT #3,GP0(JJ) : NE
XT JJ
4640    FOR JJ=0 TO GLIM : INPUT #3,GP1(JJ) : NE
XT JJ
4650    FOR JJ=0 TO GLIM : INPUT #3,GP2(JJ) : NE
XT JJ
4660    FOR JJ=0 TO GLIM : INPUT #3,GP3(JJ) : NE
XT JJ
4670    FOR JJ=0 TO GLIM : INPUT #3,GP4(JJ) : NE
XT JJ
4680    FOR JJ=0 TO GLIM : INPUT #3,GP5(JJ) : NE
XT JJ
4690    FOR JJ=0 TO GLIM : INPUT #3,GP6(JJ) : NE
XT JJ
4700    FOR JJ=0 TO GLIM : INPUT #3,GP7(JJ) : NE
XT JJ
4710    FOR JJ=0 TO GLIM : INPUT #3,GP8(JJ) : NE
XT JJ
4720    FOR JJ=0 TO GLIM : INPUT #3,GP9(JJ) : NE
XT JJ
4730   CLOSE #3 : RETURN
4740 'Error-B
4750   BEEP
4760 '*NoSave
4770           PRINT "READY" : I=I-1 : RETURN
4780 '
4790 '*KANJIR
4800           KFLAG=0
4810 '*KTOP
4820   SYMBOL (177,18),"0  1 2  3 4   5  6 7  8
9   10  1 2  3 4   5 6  7 8  9",1,1
4830   PRINT"Top Char (ｱ-ﾜ,a,/"
4840   INPUT " h,k,g,r,s0-59)";C$ : IF C$="" THE
N 4950 :'*KEXIT
4850   IK=ASC(C$) : SFLAG=0
4860   IF IK>=ASC("ｱ") AND IK<ASC("ﾜ") THEN KC=K
CODE%(IK-ASC("ｱ")+7):GOTO 5090
4870   'IF C$="4" THEN KC=KCODE%(0) : GOTO 3100
:'*KLOOP
4880   IF C$="/" THEN KC=KCODE%(1) : GOTO 5090
4890   IF C$="a" THEN KC=KCODE%(2) : GOTO 5090
4900   IF C$="h" THEN KC=KCODE%(3) : GOTO 5090
4910   IF C$="k" THEN KC=KCODE%(4) : GOTO 5090
4920   IF C$="g" THEN KC=KCODE%(5) : GOTO 5090
4930   IF C$="r" THEN KC=KCODE%(6) : GOTO 5090
4940   IF LEFT$(C$,1)="s" THEN GOTO 4990
4950 '*KEXIT
4960   LINE (100,0)-(639,31), PRESET,,BF:'erase
Kanji-Display
4970   IF KFLAG=0 THEN 4740 ELSE B(L)=-1:L=L+1:C
OLOR C,0,C: RETURN
4980 '
4990 '*KSELDM ;ﾊﾞﾝｺﾞｳ ﾆ ﾖﾙ ｶﾝｼﾞ ﾆｭｳﾘｮｸ ﾙｰﾁﾝ
5000   GOSUB 5040 :'*KSELD
5010   IF C$="n" THEN 4810 :'*KTOP
5020   SFLAG=1 : GOTO 5210 :'*KPUT
5030 '*KSELD
5040   KC=VAL(MID$(C$,2,2)) : IF KC>KCLIM THEN C
$="n" : RETURN
5050   KC=KCD(KC)
5060   XX=175 : IK=0 : LINE (175,0)-(192,16),PRE
SET,,BF
5070   PRINT@ (XX,0),KC : GOTO 5500:'*KASKOK
5080 '
5090 '*KLOOP
5100   GOSUB 5320 :'*KDISP20
5110 '*KSEL
5120   INPUT "0-19,n,b,or ret";C$
5130   IF C$=""  THEN 4810 :'*KTOP
5140   IF C$="n" OR C$="ﾐ" THEN KC=KC+20 : GOTO
5090 :'*KLoop
5150    IF C$="b" OR C$="ｺ" THEN KC=KC-20 :IF (K
C AND &HFF)>&H20 THEN 5090 ELSE KC=(KC¥&H100)*&H
100-&H94:GOTO 5090 : '*KLOOP
5160   IK=VAL(C$) : IF (IK=0) AND (C$<>"0") THEN
 5110: '*KSEL
5170   GOSUB 5430 : LINE (XX,17)-(XX+15,17),XOR,
4:'*KPOSXY,Put UnderLine
5180   GOSUB 5500 : IF C$="n" OR C$="ﾐ" THEN 511
0 :'*KASKOK
5190 '*KCOLOR
5200 '*KASKC
5210 '*KPUT
5220   INPUT "ﾖｺ ﾊﾞｲﾘﾂ(1-40) or ret";MX$
5230   IF MX$="" THEN 5260 ELSE MX=VAL(MX$)
5240   INPUT "ﾀﾃ ﾊﾞｲﾘﾂ(1-25) or ret";MY$
5250   IF MY$<>"" THEN MY=VAL(MY$)
5260   SYMBOL@ (X,Y), FNKAN$(KC+IK), MX, MY, C
5270   LINE (XX,17)-(XX+15,17),XOR,4 :'Erase Und
erLine
5280   B(L)=KC+IK : L=L+1 : GOSUB 4140: GOSUB 40
40 :'*STMXY,*STXY
5290   X=X+16*MX : GOSUB 2750 : KFLAG=1 :'*LPSET
5300   IF SFLAG=0 THEN 5110 ELSE 4810 :'*KSEL,*K
TOP
5310 '
5320 '*KDISP20
5330   GOSUB 5410: '*KERASE20
5340   IF (KC AND &HFF)>&H7E THEN KC=(KC ¥ &H100
)*&H100 +&H121
5350   XX=175 : FOR II=KC TO (KC+19)
5360   COLOR C: PRINT@ (XX,0),II: XX=XX+20
5370   IF II=KC+4 THEN XX=XX+10
5380   IF II=KC+9 THEN XX=XX+28
5390   IF II=KC+14 THEN XX=XX+10
5400   NEXT II : RETURN
5410 '*KERASE20
5420   LINE (175,0)-(639,16),PRESET,,BF: RETURN
5430 '*KPOSXX
5440   XX=175-20 : FOR II=0 TO IK
5450   XX=XX+20
5460   IF II=5 THEN XX=XX+10
5470   IF II=10 THEN XX=XX+28
5480   IF II=15 THEN XX=XX+10
5490   NEXT II : RETURN
5500 '*KASKOK
5510   INPUT "OK( NO= n, YES=CR)"; C$
5520   IF C$="n" OR C$="ﾐ" THEN C$="n": LINE (XX
,17)-(XX+15,17),XOR,4
5530   RETURN
5540 '
5550 '*KANJID
5560   KC=B(L) : L=L+1:GOSUB 4250: GOSUB 4150 :
'*GETMXY,*GETXY
5570   SYMBOL@ (X,Y),FNKAN$(KC), MX, MY, C
5580   X=X+16*MX : GOSUB 2750 : IF B(L)<>(-1) TH
EN 5550 :'*LPSET
5590   L=L+1 : RETURN
5600 '*DUM
5610 RETURN
5620 '*Error   ---------  ｴﾗｰ ｼｮﾘ ﾙｰﾁﾝ
5630          I=I-1: RESUME 1940
5640 '****************************************
************
5650 '*   PLOTTER OUTPUT
                                              *
5660 '*          CODED BY Y. MUDA, 1983.9.17
                                              *
5670 '*          RIVESED
                                              *
5680 '****************************************
************
5690   PRINT :COLOR 6
5700   INPUT "File Name";DTNM$: GOSUB 3320: '*RD
```

```
FILE
5710   INPUT "ﾊﾞｲﾘﾂ(1...5.6)";MPX
5720   INPUT "ﾀﾃｶｸﾀﾞｲ (SCREEN ﾉ ﾄｳﾘ=0, 1:1=1)";M
PY
5730    IF MPY=0 THEN MPY=MPX/.898 ELSE MPY=MPX
5740   INPUT "Color(ｸﾛ=0, ｲﾛﾂｷ=1)";COLON$ :PCOL=
1 :'1=ｸﾛ
5750   INPUT "Speed(1-10)";SP: LPRINT "T"+STR$(S
P)
5760   DIM KAF$(2): RESTORE 6620
5770   FOR II=0 TO 1: READ KAF$(II): NEXT :'READ
 KANA FONT
5780
5790   I=0:J=0:K=0:L=0
5800   GOSUB 3590 : '*GETKEYD
5810
5820   WHILE CH<>1
5830    FOR II=1 TO CMN
5840     IF CMMAND%(II)=CH THEN F=II :GOTO 5860
ELSE F=-1
5850   NEXT II
5860 ON F GOSUB 5950,5610,3660,6060,6110,6180,56
10,5610,5610,5610,5610,5610,5610,5610,5610,5610,
5610,5610,5610,5610,6260,3890,3930,5610,5550,629
0,5610,6330,5610,6010,5610,5610,5610
5870 's,0,l,b,c,->,<-,▲,▼
5880 '1 2 3 4 5 6 7 8 9 w
5890 'p m g x k a + r i . d o /
5900   GOSUB 3590 :'*GETKEYD
5910   WEND
5920   INPUT "ﾓｳｲﾁﾄﾞ ｶｻﾈｶﾞｷ ｼﾏｽｶ(Y/N)";AA$:IF AA
$="Y" OR AA$="y" THEN 5790
5930  END
5940 '
5950 '*COLORD
5960   C=B(L) : L=L+1
5970   IF C=1 THEN C=5
5980   IF C=7 THEN C=1
5990   IF C<>PCOL AND COLON$="1" THEN PCOL=C: LP
RINT "J"+STR$(C)
6000   RETURN
6010 '*PSETD
6020   GOSUB 4150 : '*GETXY
6030   LPRINT "M"+STR$(MPX*X)+","+STR$(YMAX-MPY*
Y)
6040   LPRINT "N1" :'  PLOTT POINT
6050   GOTO 2750 :'*LPSET
6060 '*LINED
6070   GOSUB 4160 : GOSUB 4180 :'*GETXY,*GETLPXY
6080   LPX$=STR$(MPX*LPX): LPY$=STR$(YMAX-MPY*LP
Y)
6090   X$=STR$(MPX*X): Y$=STR$(YMAX-MPY*Y)
6100   LPRINT "M"+LPX$+","+LPY$ : LPRINT "D"+X$+
","+Y$:GOTO 2750 :'*LPSET
6110 '*BOXD
6120   GOSUB 4160 : GOSUB 4180 :'*GETXY,*GETLPXY
6130   LPX$ =STR$(MPX*LPX) : LPY$ =STR$(YMAX-MPY
*LPY)
6140   X$=STR$(MPX*X): Y$=STR$(YMAX-MPY*Y)
6150   LPRINT "M"+LPX$+","+LPY$
6160   LPRINT "D"+X$+","+LPY$+","+X$+","+Y$+","+
LPX$+","+Y$+","+LPX$+","+LPY$:GOTO 2750 :'*LPSET
6170 '*CIRCLED
6180   RADIUS=TA(J): MFCIRC!=TA(J+1): J=J+2: GOS
UB 4150 :'*GETXY
6190   TS=0: TE=361
6200 '*CIRCJ
6210   LPRINT "M"+   STR$(MPX*(X+RADIUS*COS(TS
   *MDRF)))+","+STR$(YMAX-MPY*(Y-MFCIRC*RADIUS*S
IN(TS   *MDRF)))
6220   FOR THETA=TS TO TE STEP 6
6230     LPRINT "D"+  STR$(MPX *(X+RADIUS*COS(TH
ETA*MDRF)))+","+STR$(YMAX-MPY*(Y-MFCIRC*RADIUS*S
IN(THETA*MDRF)))
6240   NEXT THETA : GOTO 2750 :'*LPSET
6250 '*PAINTD
6260   GOSUB 4210: GOSUB 4150 :'*GETPCBC,*GETXY
6270   GOTO 2750: '*LPSET
6280 '*ARCD
6290   GOSUB 4150: RADIUS=TA(J): MFCIRC=TA(J+1):
 '*GETXY
6300   TE=(1-TA(J+2))*360: TS=(1-TA(J+3))*360: J
=J+4
6310   IF TE<TS THEN TE=TE+360
6320   GOTO 6210: '*CIRCJ
6330 '*ALPHD
6340   C$=A$(K): K=K+1: GOSUB 4250: GOSUB 4150:
'*GETMXY,*GETXY
6350   LPRINT "S"+STR$(MX*MPX) :'ALPHA-SCALE
6360   LPRINT "M"+STR$(MPX*X)+","+STR$(YMAX-MPY*
(Y+16))
6370 '       LPRINT "P"+C$
6380   FOR II2=1 TO LEN(C$)
6390    KA$=MID$(C$,II2,1)
6400    IF ASC(KA$)>&H7F THEN GOSUB 6450: GOTO 6
420: '*KANA
6410     LPRINT "P"+KA$
6420    NEXT II2
6430   RETURN
6440 '*KANA
6450   II1=1:II3=ASC(KA$)-ASC("ｱ"):MX=MX/2
6460   GOSUB 6560:'*GET KA$
6470   IF KA$="R"  THEN GOSUB 6560:DX=VAL(KA$):G
OSUB 6560:DY=VAL(KA$):LPRINT "R"+STR$(MPX*MX*DX)
+","+STR$(MPY*MX*DY): GOTO 6460
6480   IF KA$="I" THEN GOSUB 6500: GOTO 6470: 'R
ELATIVE DRAW
6490   MX=MX*2: RETURN
6500   LPRINT "I";: GOSUB 6560: DX=VAL(KA$):GOSU
B 6560:DY=VAL(KA$)
6510   LPRINT STR$(MPX*MX*DX)+","+STR$(MPY*MX*DY
);
6520   GOSUB 6560: IF KA$="R" OR KA$="E" THEN LP
RINT: RETURN :'*GET KA$
6530   DX=VAL(KA$): GOSUB 6560: DY=VAL(KA$): LPR
INT ",";: GOTO 6510
6540 '
6550 '*GET NEXT CHRS INTO KA$ FROM KAF$
6560   KA$=MID$(KAF$(II3), II1, 1): II1=II1+1: I
F KA$=" " THEN 6560
6570   IF KA$="E" OR KA$="" THEN RETURN
6580   CH$=MID$(KAF$(II3), II1, 1): II1=II1+1
6590   IF CH$="," THEN RETURN
6600   IF CH$=" " THEN 6580 ELSE KA$=KA$+CH$: GO
TO 6580
6610 '*KANA FONT
6620 DATA"R,1,14,I,10,0,0,-3,-2,-4,R,-3,3,I,0,-4
,-2,-3,-3,-2,R,15,0,E:'ｱ
6630 DATA"R,0,5,I,6,4,5,4,R,-5,-4,I,0,-9,R,10,0,
E                   :'ｲ
6640 '.SAVE"PenPlt-2
```

RANDOM BOX

Kコンパイラのゲームを FM-11で

■COMPAC K

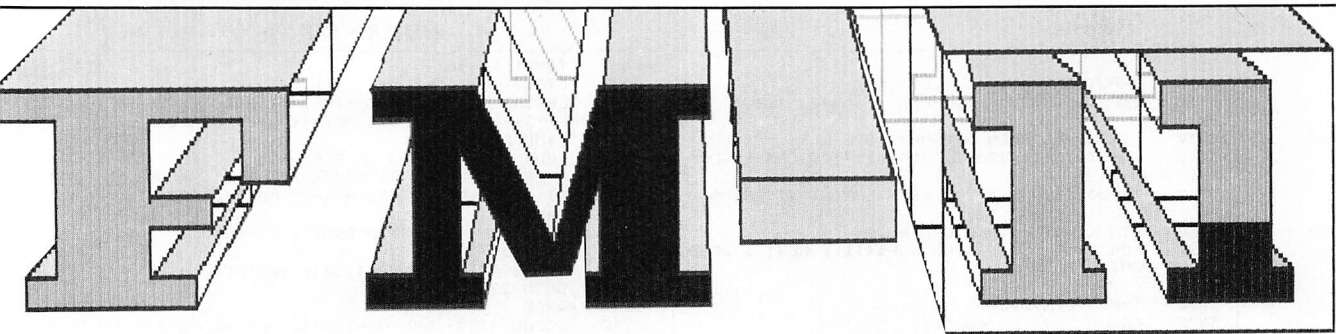
現在までに発表されているKコンパイラ用グラフィック・サブルーチンなどはFM-7/8用なので，サブルーチンによってはFM-11用に変更しなければなりません．この変更は，KEYサブルーチン以外，グラフィック画面の初期化，引数の変更などです．

参考に“MELON PANIC”の変更箇所をリスト1に示します．その他，410行，PSG，PSG1，MUS，ONGAなどのサウンド関係をすべて取ります．そして，マシン語は＄701B～＄78FFまでを入力します．

サブシステムのコマンドについては，資料編を参照してください．

リスト1

```
1510 'WID;HLT[];ADDR=$FC82;CPOKE ADDR ,$1,0,40,25,0,25,0,1,0;SUB[]
1511 '    HLT[];ADDR=$FC82;CPOKE ADDR,$21,$1F;SUB[]
1512 '    HLT[];ADDR=$FC82;CPOKE ADDR,$22,$1F;SUB[]
1513 '    HLT[];ADDR=$FC82;CPOKE ADDR,$25,0;SUB[];RETURN
1550 'SUB;CODE $7F,$FD,5,$B6,$FD,$05,$2B,$F8,$39
1570 'INK;HLT[];ADDR=$FC82;CPOKE ADDR,$2D;SUB[];HLT[];KYCH=PEEK($FC83);KY=PEEK($
FC84);POKE $FC82,$3F;SUB[];IF KYCH=$80 THEN KY=KY1 ;FI;KY=KY;KY1=KY;RETURN
```

F-BASIC, CP/M-80

フルカラーグラフィック画面ハードコピー

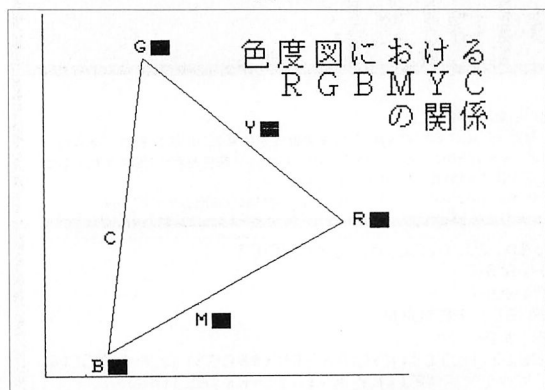
■六田嘉明
細川治雄

'83年3月に発売になったシャープのカラープリンタMZIP04を使って,FM-7/11のグラフィック画面のカラーハードコピーをとるプログラムを作成しましたので,発表します.

FM-7の画面は640×200ドット,FM-11は640×200と640×400の二種類があります.また,MZ1P04は横が10インチ,1024ドットで各点ごとに8色のカラーを印刷できます.そこで画面モードが640×400ドットの場合で画面とプリンタのドットが1対1に対応する普通版(CCOPY)と,縦,横とも2倍に拡大する拡大版(WCCOPY)の2種類,作りました.また,640×200ドットの場合には,縦を2倍に拡大して400ラインとしてプリントしています.

使い方

FMシリーズではグラフィック画面とテキスト画面が別画面として扱えるので,ハードコピーを取りたい画面でプログラムをブレークさせて,今回のカラーハード・コピープログラムを**LOAD**して**RUN**します.**RUN**後,画面とハードコピー間の白黒反転の有無(BKTYP 0 OR 1)および画面のドットモード(640×400(0)か,640×200(1)か)を入力することにより動作が開始します.

このプログラムはマシン語もBASICで入力するようにしてあるので,あなたのBASICプログラム中に本プログラムをマージして,**CLEAR**文を先頭に移し,かつ600または610行目の**END**文を**RETURN**文にかえてサブルーチンにすれば,プログラムを壊してしまうこともなくなります.

使い方で注意してほしいのは,FM-7の場合は画面のドット・モードは640×200しか選べないこと,7/11ともマルチ・ページ画面を持っているので,もし裏画面がCRT上に表示されている場合は,ダイレクト命令で,**SCREEN n,n**を入力しなければならないという点です(nは画面番号).

なおハード・コピーの寸法および所要時間は,**表1**に示してあります.普通版で,はがきより一回り小さい程度,拡大版では12インチのブラウン管より少し大きい程度のコピーが取れます.拡大版でも極めてきめの細かい,ハードコピーが得られます.

表1 ハードコピーの寸法および時間(F-BASIC版)

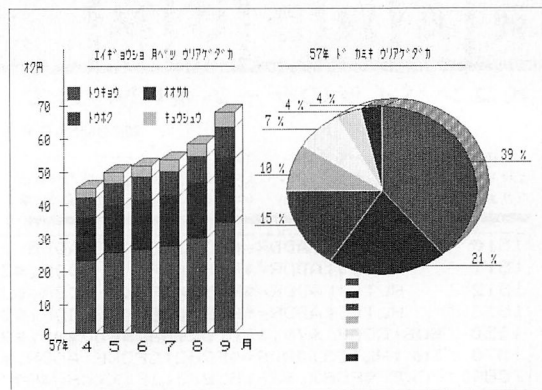
		普通版(CCOPY)	拡大版(WCCOPY)
大きさ(mm)		84×134.5	168×269
時間	640×400	3分00秒程度	12分程度
	640×200	2分40秒程度	11分程度

光の三原色から色の三原色への変換について

みなさんも中学校の美術の授業などで習ったと思いますが,光の三原色はレッド(R),グリーン(G),ブルー(B)であり,絵の具の三原色はマゼンタ(紅,M),イエロー(黄,Y),シアン(青,C)となっています.本プリンタでも,M・Y・Cの組合せで7色を合成しています(ただし,黒はインクがにじむので黒インクで,表わすことになっています).

そこで今回はRGBで表わされている色を,MYCと黒

図1

図2 FM-11のデモのハードコピー

(BK)の4色を使って表わすための関係を話します．

RGBとMYCの関係は，**図1**の色相表のようになっています．たとえばマゼンタ(紅，M)はRとBを混ぜたものですから，RまたはBの一方または双方が1(ON)で，かつGが0(OFF)であるとき，Mを1(ON)にするという論理で与えます．これは，次のような論理式で表現できます．

マゼンタ：M＝ (R OR B) AND $\bar{G}$

ここで$\bar{G}$はGの否定(NOT G)を意味します．

同様にして，イエロー(Y)とシアン(C)の論理式は，

イエロー：Y＝ (R OR G) AND $\bar{B}$ シアン： C＝ (G OR B) AND $\bar{R}$

となります．

次に，ブラック(BK)はCRT上の黒(空白)をプリンタ上でも黒で表現する場合(BKTYP 0)と，逆に空白で表現する場合(BKTYP 1)の2つを選択できるようにしました．

先の場合は，CRT上でRGBがすべて0(空白)の部分をインクの黒で表現するので，論理式は，

ブラック：BK＝NOT(R OR G OR B)

となり，後の場合はRGBがすべて1(白)の部分を黒で表現するので，

ブラック：BK＝$\bar{R}$ AND $\bar{G}$ AND $\bar{B}$

となります．

R, G, B, M, Y, CのFM-7/11でのカラーレコードおよびプリンタにおけるカラーコードの対応表を**表2**に示しておきます．たとえば，FMでの紫はMZ1P04ではマゼンダ(紅)に，FMでの青はMZでの紫に，FMでの水色はMZでのシアン(青)に対応していて，これでほぼ，画面とコピーの色が合います．画面の青はあざやかな青ですが，プリンタでは紫(深い青)になる点だけ異なります．しかしこれは，これ以上どうしようもありません．

表2 カラーコード対応表

FM7 11		MZ1P04	
カラーコード	色	カラーコード	色
0	黒	1	黒
1	青	7	紫 (紅＋青)
2	赤	3	赤 (紅＋黄)
3	紫(赤＋青)	2	紅〔マゼンタ〕
4	緑	5	緑 (黄＋青)
5	水色(緑＋青)	6	青〔シアン〕
6	黄(赤＋緑)	4	黄〔イエロー〕
7	白	0	無印字

F-BASIC版について

FMシリーズでは，CRTの制御はすべてディスプレイ・サブシステムのCPUが行なっているため，ドット・イメージ・データの読み出しは，BIOSのRCBインターフェイスのSUBINコマンド中のGET-BLOCK 2サブコマンドを用いて行なっています．CRT上で読み出したい領域の対角線のX，Y座標を送ることにより，RGBのドットイメージ・データが指定したバッファ・エリア(RCBDB)に読み込まれます．

一方，プリンタにドット・イメージを描かせる命令は ESC ＋ I ＋ n ＋ m ＋ x_1, x_2… x_m で，nは色とライン番号を表わす1バイトのASCII表現，Mは転送バイト数，X_1……X_mは，ドット・イメージ・データです．

そこで，ディスプレイ・サブシステムから取り出したRGBデータから先に述べた論理によって，マゼンタ，イエロー，シアン，ブラックのデータを作り，プリンタの1ライン分のデータを作り，プリンタの1ライン分のデータを，X_1，…X_mとします．

これをBIOSのRCBインターフェイスのLPOUTコマンドを使って，プリンタに送ります．そして4ライン分のデータを送った後，一回マイクロ・ラインフィード命令 ESC ＋Ⓐをプリンタへ送ると，4ライン分がプリンタに描かれるわけです．

なお拡大版の場合には，各ビットを2ビットに拡大操作してプリンタへのデータとしています．

またこのプログラムのマシン語部分は，リロケータブルに作ってあるので，**表3**の部分を手直しすれば，ユーザーエリアの任意の場所に移すことができます．

SHARP MZ1P04はFMシリーズとASCIIコードが一部違っているために，グラフィック・キャラクタおよびカタカナの〝，〟，〝．〟，〝ー〟の3つと〝¥〟が正しくプリントされないのは，残念です．しかしながら，このような美しいハードコピーも取れ，通常のリスト・アウトもできる，すぐれたプリンタであることは確かです．

表3 マシン語部分を移動する場合の変更箇所

CCOPY			WCCOPY		
行番号	旧データ	新データ	行番号	旧データ	新データ
90	6FFF	6FFF±α	90	6FFF	6FFF±α
130	7000	7000 ±α	130	7000	7000 ±α
820	7010	7010 ±α	810	7010	7010 ±α
880	7020	7020±α	870	7020	7020 ±α
920	70B0	70B0 ±α	920	7080	7080 ±α
980	70C0	70C0±α	980	7090	7090 ±α
1020	7024	7024 ±α	1030	7024	7024 ±α
1610	72B8	72B8 ±α	2220	7248	7248 ±α
1660	72BE	72BE ±α	2270	724E	724E ±α

CP/M-80版

プログラムの入力方法

まず，CP/Mを起動した後（**Z80カードが必要です**），DDT (8080ディバッガ) をRUNさせます．つぎに，念のため，**F** (Fill) コマンドで，100Hから0FFFHまでをクリアしておきます (**F100, FFF, 00**)．そして，S(Set) コマンドで，ダンプ・リスト（**リスト3，4**）の内容を，100H番地から入力してください．入力が終われば，**D**(Dump) コマンドで確認した後，一度CP/Mに戻ります(**G0**または CTRL ＋C)．直ちに（内容が壊れないうちに)，CP/Mのビルトイン・コマンドのSaveを使って，ディスクにセーブします(**SAVE 4 CCOPY. COM**または，**SAVE 4 WCCOPY. COM**)．

以後，CCOPY，またはWCCOPYというトランジェント・コマンドで，グラフィック画面のカラーハードコピーがとれるようになります．

所要時間はオール・マシン語のため，若干速まり，CCOPYで2分30秒程度，WCCOPYで9分程度でプリントできます．

終わりに

サンプルを見るとわかると思いますが，びっくりするほど美しいカラーハードコピーが取れます．

初めオールBASICで作ったと時には，普通版で40分もかかったので，主要な部分をマシン語に置き換えたわけです．6809のアセンブラがなかったために，ハンド・アセンブルを行なうこととなり，苦労しました．またFM-11を使って作ったのですが，まだ完全なマニュアルが完成していなかったので，FM-7の資料を使って作成することになってしまいました．しかしこうして完成して，美しいハードコピーが出てくると，何かワクワクしてきます．

FM-8では，BIOSのアドレスが少し違うだけだと思いますので，簡単に移植できると思います．

最後に，このプログラムで，きっとグラィックの楽しみが倍増されると思います．

リスト1　F-BASIC用　CCOPY

```
10 '*****************************************************
20 '*             COLOR HARD COPY ==CCOPY==             *
30 '*      (SHARP カラ█インクジ゛ェット フ゜リンタ█ MZ-1P04)     *
40 '*                High Speed Version                 *
50 '*                                                   *
60 '*      1983.4.8 CODED BY Y. MUDA & H.HOSOKAWA       *
70 '*                   4.16 カンセイ !                    *
80 '*****************************************************
90        CLEAR ,&H6FFF
100       LPRINT CHR$(27);"W09";
110       LPRINT CHR$(27);"H001105";
120 '
130       BIOS%=&H7000:RCB%=BIOS%+&H10:RCBDB%=BIOS%+&H20:BLU%=BIOS%+&H24
140       RED%=BIOS%+&H4C:GRN%=BIOS%+&H74
150       BIOSLPO%=BIOS%+&HA0:RCBLPO%=BIOS%+&HB0:BK%=BIOS%+&HC0
160       MAZ%=BIOS%+&H116:YEL%=BIOS%+&H16C:CIA%=BIOS%+&H1C2
170       RGBMYC%=BIOS%+&H220: BIOSMLF%=BIOS%+&H2B0
180 '
190       DEF USR0=BIOS% :DEF USR1=BIOSLPO% :DEF USR2=RGBMYC%:DEF USR3=BIOSMLF%
200 '
210      FOR I%=0 TO &H17:READ A$:POKE BIOS%+I%    ,VAL("&H"+A$):NEXT
220      FOR I%=0 TO &H15:READ A$:POKE BIOSLPO%+I%,VAL("&H"+A$):NEXT
230      FOR I%=0 TO &H85:READ A$:POKE RGBMYC%+I% ,VAL("&H"+A$):NEXT
240      FOR I%=0 TO &H0F:READ A$:POKE BIOSMLF%+I%,VAL("&H"+A$):NEXT
250      FOR I%=0 TO 3
260        POKE BK%+(86*I%)+0,&H1B
270        POKE BK%+(86*I%)+1,&H49
280        POKE BK%+(86*I%)+3,&H30
290        POKE BK%+(86*I%)+4,&H38
300        POKE BK%+(86*I%)+5,&H30
310      NEXT
320      ESC$=CHR$(&H1B)
330      COUNT=0: Y1=0: Y2=0
340 '
350       INPUT "DISPRAYノ シロヲ PRINTERテ゛ シロ=0 ,クロ=1";BKTYP%
360       INPUT "DISPRAYモ█ド゛█640*400=0,640*200=1";DH%: IF DH%<0 OR DH%>1 THEN 360
370      LPRINT ESC$;CHR$(&H18);: 'Clear Printer Buffer
380 '
390 '*LPA
400      LN%=0
410 '*LPB
420        X1=0: X2=319
430        GOSUB 650 : 'Call *GETSCRN
440        MID%=0:C%=USR2(MID%+BKTYP%*2) : 'RGB~to~BMYC
450        X1=320 : X2=639
460        GOSUB 650 : 'Call *GETSCRN
470        MID%=1:C%=USR2(MID%+BKTYP%*2) : 'RGB~to~BMYC
480        FOR II%=0 TO DH%
490          FOR I%=0 TO 3 :POKE BK%+(86*I%)+2 ,LN%+&H30+(4*I%):NEXT I%
500          B%=USR1(0)       :'CALL BIOSLPO
510          LN%=LN%+1
520        NEXT II%
530        Y1=Y1+1: Y2=Y1
540        IF LN% <> 4 THEN 420 :'*LPB
550      ' *END OF LPB LOOP
560      '
570      B%=USR3(0)            :'LPRINT ESC$;"A";  (SEND MICRO LF)
580      COUNT=COUNT+1
590      LPRINT ESC$;CHR$(&H18);  : 'Clear Printer Buffer
600      IF COUNT <> 100 THEN 400 :'*LPA
610      END
620 '
630 '
640 '*GETSCRN (Get Screen Dot Image into RCBDB=(BLU%)
650      POKE RCB%  + 4, 0 :'Write RCBLNH=11 バイト
660      POKE RCB%  + 5, &H0B
670      POKE RCBDB%+ 2, &H1D            :'GET~BLOCK~2 COMMAND CODE
680      POKE RCBDB%+ 3, X1 \   &H100    :'Write X1
690      POKE RCBDB%+ 4, X1 MOD &H100
700      POKE RCBDB%+ 5, Y1 \   &H100    :'Write Y1
710      POKE RCBDB%+ 6, Y1 MOD &H100
```

リスト１　F-BASIC用　CCOPY

```
720      POKE RCBDB%+ 7, X2 \  &H100       :'Write X2
730      POKE RCBDB%+ 8, X2 MOD &H100
740      POKE RCBDB%+ 9, Y2 \  &H100       :'Write Y2
750      POKE RCBDB%+10, Y2 MOD &H100
760      A%=USR0(0)                        :'Call BIOS
770      RETURN
780 '
790 '
800 '
810 '
820  DATA 8E,70,10         :'7000      LDX #RCBAD
830  DATA AD,9F,FB,FA      :'7003      JSR [$FBFA]=BIOS
840  DATA 39               :'7007      RTS
850  DATA 0,0,0,0,0,0,0,0:'7008
860  DATA 11               :'7010 RCB: SUBIN CMD
870  DATA 0                :'7011      don't care
880  DATA 70,20            :'7012      RCBDBA=$7020
890  DATA 00,0B            :'7014      RCBLNH=11
900  DATA 00,7c            :'7016      RCBBMH=120(=640/2/8 *3)+4
910 '
920  DATA 8E,70,B0          :'70A0  LDX #RCBLPOAD
930  DATA AD,9F,FB,FA       :'70A3  JSR [$FBFA]=BIOS
940  DATA 39                :'70A7  RTS
950  DATA 0,0,0,0,0,0,0,0 :'70A8
960  DATA 0E                :'70B0  RCBLPO :LPOUT CMD
970  DATA 0                 :'70B1         DON`T CARE
980  DATA 70,C0             :'70B2         RCBDBA=$70C0
990  DATA 01,58             :'7084         RCBLNH=86*4=&H0158
1000 '
1010 DATA A6,03           :'7220 RGBMYC :  LDA 3,X          ;A=INPUT PARAMATOR
1020 DATA 8E,70,24        :'7222           LDX #$7024       ;X=BLU% ADRESS
1030 DATA 33,89,01,F4     :'7225           LEAU $01F4,X     ;U=WORKING ARIA ADRESS
1040 DATA 31,89,00,A2     :'7229           LEAY $00A2,X     ;Y=BKDT% ADRESS
1050 DATA AF,C4           :'722D           STX  ,U          ;STORE BLU% TO WORK.ARA
1060 DATA 85,01           :'722F           BITA #$01        ;MID%=0 ?
1070 DATA 27,03           :'7231           BEQ P1           ;YES THEN JMP P1
1080 DATA 31,A8,28        :'7233           LEAY $28,Y       ;NO THEN BKDT=BKDT%+40
1090 DATA 10,AF,42        :'7236 P1     :  STY 2,U          ;STORE BKDT TO WORK.ARA
1100 DATA C6,28           :'7239           LDB #$28         ;B=40 (SET COUNTOR)
1110 DATA 85,02           :'723B           BITA #$2         ;BKTYP%=0 ?
1120 DATA 26,13           :'723D           BNE BKT1         ;NO THEN JMP BKT1
1130 DATA A6,84           :'723F BKT0   :  LDA  ,X          ;BLU
1140 DATA AA,88,28        :'7241           ORA $28,X        ;OR RED
1150 DATA AA,88,50        :'7244           ORA $50,X        ;OR GRN
1160 DATA 43              :'7247           COMA             ;NOT(BLU OR RED OR GRN)
1170 DATA A7,A0           :'7248           STA  ,Y+         ;STORE BLACK,AUTO INC Y
1180 DATA A6,80           :'724A           LDA  ,X+         ;AUTO INC X
1190 DATA 5A              :'724C           DECB             ;B=B-1
1200 DATA 26,F0           :'724D           BNE BKT0         ;B>0 THEN JMP BKT0
1210 DATA 6E,CC,0F        :'724F           JMP +$0F,PC      ;JMP MYC
1220 DATA A6,84           :'7252 BKT1   :  LDA  ,X          ;BLU
1230 DATA A4,88,28        :'7254           ANDA $28,X       ;AND RED
1240 DATA A4,88,50        :'7257           ANDA $50,X       ;AND GRN
1250 DATA A7,A0           :'725A           STA  ,Y+         ;STORE BLACK,AUTO INC Y
1260 DATA A6,80           :'725C           LDA  ,X+         ;AUTO INC X
1270 DATA 5A              :'725E           DECB             ;B=B-1
1280 DATA 26,F1           :'725F           BNE BKT1         ;B>0 THEN JMP BKT1
1290 DATA AE,C4           :'7261 MYC    :  LDX  ,U          ;X=BLU% ADRESS
1300 DATA 10,AE,42        :'7263           LDY 2,U          ;Y=BKDT%
1310 DATA C6,28           :'7266           LDB #$28         ;B=40 (SET COUNTOR)
1320 DATA 34,04           :'7268 MYCLOOP:  PSHS B           ;COUNTOR STORE
1330 DATA A6,84           :'726A           LDA  ,X          ;BLU
1340 DATA AA,88,28        :'726C           ORA $28,X        ;OR RED
1350 DATA E6,88,50        :'726F           LDB $50,X        ;GRN
1360 DATA 53              :'7272           COMB             ;NOT (GRN)
1370 DATA E7,44           :'7273           STB 4,U          ;STORE NOT(GRN) TO W.A
1380 DATA A4,44           :'7275           ANDA 4,U         ;AND NOT(GRN)
1390 DATA A7,A8,56        :'7277           STA $56,Y        ;STORE MAZNTA
1400 DATA A6,88,28        :'727A           LDA $28,X        ;RED
1410 DATA AA,88,50        :'727D           ORA $50,X        ;OR GRN
1420 DATA E6,84           :'7280           LDB  ,X          ;BLU
1430 DATA 53              :'7282           COMB             ;NOT (BLU)
1440 DATA E7,45           :'7283           STB 5,U          ;STORE NOT(BLU)
1450 DATA A4,45           :'7285           ANDA 5,U         ;AND NOT(BLU)
1460 DATA A7,A9,00,AC     :'7287           STA $00AC,Y      ;STORE YELLOW
1470 DATA A6,84           :'728B           LDA  ,X          ;BLU
1480 DATA AA,88,50        :'728D           ORA $50,X        ;OR GRN
1490 DATA E6,88,28        :'7290           LDB $28,X        ;RED
1500 DATA 53              :'7293           COMB             ;NOT (RED)
1510 DATA E7,46           :'7294           STB 6,U          ;STORE NOT(RED)TO W.A
1520 DATA A4,46           :'7296           ANDA 6,U         ;AND NOT(RED)
1530 DATA A7,A9,01,02     :'7298           STA $0102,Y      ;STORE CIAN
1540 DATA A6,80           :'729C           LDA  ,X+         ;AUTO INC X
1550 DATA A6,A0           :'729E           LDA  ,Y+         ;AUTO INC Y
1560 DATA 35,04           :'72A0           PULS B           ;PULL B (COUNTOR)
1570 DATA 5A              :'72A2           DECB             ;B=B-1
```

リスト1　F-BASIC用　CCOPY

```
1580 DATA 26,C3          :'72A3            BNE MYCLOOP     ;B>0 THEN JMP MYCLOOP
1590 DATA 39             :'72A5            RTS
1600 '
1610 DATA 8E,72,B8       :'72B0            LDX #RCBMLF
1620 DATA AD,9F,FB,FA    :'72B3            JSR [$FBFA]
1630 DATA 39             :'72B7            RTS
1640 DATA 0E             :'72B8 RCBMLF:    LPOUT CMD
1650 DATA 0              :'72B9            DON'T CARE
1660 DATA 72,BE          :'72BA            RCBDBA=$72BE
1670 DATA 00,02          :'72BC            RCBLNH=2
1680 DATA 1B,41          :'72BE RCBMLFDB:ESC,"A"
1690 '********* END OF PROGRAM CCOPY *********
1700 ' SAVE "CCOPY"
****************** ﾁｭｳｲ **********************
*           ｳｴﾉ LIST ﾁｭｳﾉ   ■, ■, \  ﾊ          *
*            FM 7,11ﾃﾞﾊ     -, ., ¥  ﾃﾞｽ        *
*********************************************
```

注：360行　モード␣640

リスト2　F-BASIC用　WCCOPY

```
10 '*********************************************
20 '*          WIDE COLOR HARD COPY ==WCCOPY==     *
30 '*       (SHARP ｶﾗ■ｲﾝｸｼﾞｪｯﾄ ﾌﾟﾘﾝﾀ■ MZ-1P04)   *
40 '*             High Speed Version               *
50 '*                                              *
60 '*      1983.4.8 CODED BY Y. MUDA & H.HOSOKAWA  *
70 '*                4.22 ｶﾝｾｲ !                   *
80 '*********************************************
90      CLEAR ,&H6FFF
100     LPRINT CHR$(27);"W09";
110     LPRINT CHR$(27);"H001105";
120 '
130     BIOS%=&H7000:RCB%=BIOS%+&H10:RCBDB%=BIOS%+&H20:BLU%=BIOS%+&H24
140     RED%=BIOS%+&H3D:GRN%=BIOS%+&H56
150     BIOSLPO%=BIOS%+&H70:RCBLPO%=BIOS%+&H80:BK%=BIOS%+&H90
160     MAZ%=BIOS%+&HFA:YEL%=BIOS%+&H164:CIA%=BIOS%+&H1CE
170     RGBMYC%=BIOS%+&H250:BIOSMLF%=BIOS%+&H240
180 '
190     DEF USR0=BIOS% :DEF USR1=BIOSLPO% :DEF USR2=RGBMYC%:DEF USR3=BIOSMLF%
200 '
210 RESTORE 810:FOR I%=0 TO &H17:READ A$:POKE BIOS%+I%    ,VAL("&H"+A$):NEXT
220 RESTORE 920:FOR I%=0 TO &H15:READ A$:POKE BIOSLPO%+I%,VAL("&H"+A$):NEXT
230 RESTORE1020:FOR I%=0 TO &HFD:READ A$:POKE RGBMYC%+I% ,VAL("&H"+A$):NEXT
240 RESTORE2220:FOR I%=0 TO &H0F:READ A$:POKE BIOSMLF%+I%,VAL("&H"+A$):NEXT
250     FOR I%=0 TO 3
260       POKE BK%+(106*I%)+0,&H1B
270       POKE BK%+(106*I%)+1,&H49
280       POKE BK%+(106*I%)+3,&H31
290       POKE BK%+(106*I%)+4,&H30
300       POKE BK%+(106*I%)+5,&H30
310     NEXT
320     COUNT=0: X1=0: X2=0
330 '
340     INPUT "DISPRAYﾉ ｼﾛｦ PRINTERﾃﾞ ｼﾛ=0 ,ｸﾛ=1";BKTYP%
350     INPUT "DISPRAYﾓ■ﾄﾞ■640*400=0,640*200=1";DH%:IF DH%<0 OR DH%>1 THEN 350
360     LPRINT ESC$;CHR$(&H18);: 'Clear Printer Buffer
370 '
380 '*LPA
390     LN%=0
400 '*LPB
410       Y1=0: Y2=199
420       GOSUB 640 : 'Call *GETSCRN
430       MID%=0:C%=USR2(MID%+BKTYP%*2+DH%*4) : 'RGB to BMYC
440       IF DH%=1 THEN 480
450         Y1=200 : Y2=399
460         GOSUB 640 : 'Call *GETSCRN
470         MID%=1:C%=USR2(MID%+BKTYP%*2+DH%*4) : 'RGB to BMYC
480       FOR II%=0 TO 1
490         FOR I%=0 TO 3 :POKE BK%+(106*I%)+2 ,LN%+&H30+(4*I%):NEXT I%
500         B%=USR1(0)      :'CALL BIOSLPO
510         LN%=LN%+1
520       NEXT II%
530       X1=X1+1: X2=X1
540       IF LN% <> 4 THEN 410 :'*LPB
550     ' *END OF LPB LOOP
560     '
570     B%=USR3(0)          :'LPRINT ESC$;"A";  (SEND MICRO LF)
580     COUNT=COUNT+1
590     IF COUNT <> 320 THEN 390 :'*LPA
600     END
610 '
620 '
630 '*GETSCRN (Get Screen Dot Image into RCBDB=(BLU%)
640     POKE RCB%  + 4, 0 :'Write RCBLNH=11 ﾊﾞｲﾄ
650     POKE RCB%  + 5, &H0B
```

```
660      POKE RCBDB%+ 2, &H1D              :'GET~BLOCK~2 COMMAND CODE
670      POKE RCBDB%+ 3, X1 \   &H100      :'Write X1
680      POKE RCBDB%+ 4, X1 MOD &H100
690      POKE RCBDB%+ 5, Y1 \   &H100      :'Write Y1
700      POKE RCBDB%+ 6, Y1 MOD &H100
710      POKE RCBDB%+ 7, X2 \   &H100      :'Write X2
720      POKE RCBDB%+ 8, X2 MOD &H100
730      POKE RCBDB%+ 9, Y2 \   &H100      :'Write Y2
740      POKE RCBDB%+10, Y2 MOD &H100
750      A%=USR0(0)                        :'Call BIOS
760      RETURN
770 '
780 '
790 '
800 '
810 DATA 8E,70,10          :'7000          LDX #RCBAD
820 DATA AD,9F,FB,FA       :'7003          JSR [$FBFA]=BIOS
830 DATA 39                :'7007          RTS
840 DATA 0,0,0,0,0,0,0,0:'7008
850 DATA 11                :'7010 RCB:     SUBIN CMD
860 DATA 0                 :'7011          don't care
870 DATA 70,20             :'7012          RCBDBA=$7020
880 DATA 00,0B             :'7014          RCBLNH=11
890 DATA 00,4F             :'7016          RCBBMH=75(=400/2/8 *3)+4
900 '
910 '
920 DATA 8E,70,80          :'7070          LDX #RCBLPOAD
930 DATA AD,9F,FB,FA       :'7073          JSR [$FBFA]=BIOS
940 DATA 39                :'7077          RTS
950 DATA 0,0,0,0,0,0,0,0:'7078
960 DATA 0E                :'7080 RCBLPO:  LPOUT CMD
970 DATA 0                 :'7081          DON'T CARE
980 DATA 70,90             :'7082          RCBDBA=$7090
990 DATA 01,A8            :'7084          RCBLNH=106*4=&H01A8
1000 '
1010 '
1020 DATA A6,03            :'7250 RGBMYC : LDA 3,X          ;A=INPUT PARAMATOR
1030 DATA 8E,70,24         :'7252          LDX #$7024       ;X=BLU% ADRESS
1040 DATA 33,89,02,14      :'7255          LEAU $0214,X     ;U=WORKING ARIA ADRESS
1050 DATA 31,89,00,D5      :'7259          LEAY $00D5,X     ;Y=BKDTEND ADRESS
1060 DATA AF,C4            :'725D          STX  ,U          ;STORE BLU% TO WORK.ARA
1070 DATA 85,01            :'725F          BITA #$01        ;MID%=0 ?
1080 DATA 27,03            :'7261          BEQ P1           ;YES THEN JMP P1
1090 DATA 31,A8,CE         :'7263          LEAY-$32,Y       ;NO THEN BKDT=BKDT%-50
1100 DATA 10,AF,42         :'7266 P1     : STY 2,U          ;STORE BKDT TO WORK.ARA
1110 DATA C6,19            :'7269          LDB #$19         ;B=25 (SET COUNTOR)
1120 DATA A7,47            :'756B          STA 7,U          ;STORE INPPARAM TO W.A
1130 DATA 85,02            :'726D          BITA #$2         ;BKTYP%=0 ?
1140 DATA 26,18            :'726F          BNE BKT1
1150 DATA 34,04            :'7271 BKT0   : PSHS B           ;PUSH B(COUNTOR)
1160 DATA A6,84            :'7273          LDA  ,X          ;BLU
1170 DATA AA,88,19         :'7275          0RA $19,X        ;OR RED
1180 DATA AA,88,32         :'7278          ORA $32,X        ;OR GRN
1190 DATA 43               :'727B          COMA             ;NOT(BLU OR RED OR GRN)
1200 DATA 17,00,85         :'727C          LBSR BITEXTD
1210 DATA A6,80            :'727F P2     : LDA  ,X+         ;AUTO INC X
1220 DATA 35,04            :'7281          PULS B           ;PULL B(COUNTOR)
1230 DATA 5A               :'7283          DECB             ;B=B-1
1240 DATA 26,EB            :'7284          BNE BKT0
1250 DATA 6E,CC,13         :'7286          JMP +$13,PC      ;JMP MYC
1260 DATA 34,04            :'7289 BKT1   : PSHS B           ;PUSH B(COUNTOR)
1270 DATA A6,84            :'728B          LDA  ,X          ;BLU
1280 DATA A4,88,19         :'728D          ANDA $19,X       ;AND RED
1290 DATA A4,88,32         :'7290          ANDA $32,X       ;AND GRN
1300 DATA 8D,6F            :'7293          BSR BITEXTD
1310 DATA A6,80            :'7295 P3     : LDA  ,X+         ;AUTO INC X
1320 DATA 35,04            :'7297          PULS B           ;PULL B(COUNTOR)
1330 DATA 5A               :'7299          DECB             ;B=B-1
1340 DATA 26,ED            :'729A          BNE BKT1         ;B>0 THEN JMP BKT1
1350 DATA AE,C4            :'729C MYC    : LDX  ,U          ;X=BLU% ADRESS
1360 DATA 10,AE,42         :'729E          LDY 2,U          ;Y=BKDTEND ADRESS
1370 DATA C6,19            :'72A1          LDB #$19         ;B=25 (SET COUNTOR)
1380 DATA 34,04            :'72A3 MYCLOOP: PSHS B           ;COUNTOR STORE
1390 DATA A6,84            :'72A5          LDA  ,X          ;BLU
1400 DATA AA,88,19         :'72A7          ORA $19,X        ;OR RED
1410 DATA E6,88,32         :'72AA          LDB $32,X        ;GRN
1420 DATA 53               :'72AD          COMB             ;NOT (GRN)
1430 DATA E7,44            :'72AE          STB 4,U          ;STORE NOT(GRN) TO W.A
1440 DATA A4,44            :'72B0          ANDA 4,U         ;AND NOT(GRN)
1450 DATA 34,20            :'72B2          PSHS Y           ;PUSH DATA ADRESS
1460 DATA 31,A9,00,6A      :'72B4          LEAY +$006A,Y    ;MAKE MAZDT ADRESS
1470 DATA 8D,4A            :'72B8          BSR BITEXTD
1480 DATA 35,20            :'72BA          PULS Y           ;PULL DATA ADRESS
1490 DATA A6,88,19         :'72BC P4     : LDA $19,X        ;RED
1500 DATA AA,88,32         :'72BF          ORA $32,X        ;OR GRN
1510 DATA E6,84            :'72C2          LDB  ,X          ;BLU
```

リスト2　F-BASIC用　WCCOPY

```
1520 DATA 53              :'72C4            COMB           ;NOT (BLU)
1530 DATA E7,45           :'72C5            STB 5,U        ;STORE NOT(BLU)
1540 DATA A4,45           :'72C7            ANDA 5,U       ;AND NOT(BLU)
1550 DATA 34,20           :'72C9            PSHS Y         ;PUSH DATA ADRESS
1560 DATA 31,A9,00,D4     :'72CB            LEAY +$00D4,Y  ;MAKE YELDATA ADRESS
1570 DATA 8D,33           :'72CF            BSR BITEXTD
1580 DATA 35,20           :'72D1            PULS Y         ;PULL DATA ADRESS
1590 DATA A6,84           :'72D3 P5      :  LDA  ,X        ;BLU
1600 DATA AA,88,32        :'72D5            ORA $32,X      ;OR GRN
1610 DATA E6,88,19        :'72D8            LDB $19,X      ;RED
1620 DATA 53              :'72DB            COMB           ;NOT (RED)
1630 DATA E7,46           :'72DC            STB 6,U        ;STORE NOT(RED)TO W.A
1640 DATA A4,46           :'72DE            ANDA 6,U       ;AND NOT(RED)
1650 DATA 34,20           :'72E0            PSHS Y         ;PUSH DATA ADRESS
1660 DATA 31,A9,01,3E     :'72E2            LEAY +$013E,Y  ;MAKE CIAN ADRESS
1670 DATA 8D,1C           :'72E6            BSR BITEXTD
1680 DATA 35,20           :'72E8            PULS Y         ;PULL DATA ADRESS
1690 DATA E6,47           :'72EA            LDB 7,U        ;LD INPUTPARAMATOR
1700 DATA C5,04           :'72EC            BITB #$04      ;DH%=0?
1710 DATA 27,02           :'72EE            BEQ P6         ;YES THEN JMP P6
1720 DATA A6,A3           :'72F0            LDA  ,Y--      ;AUTO 2 DEC Y
1730 DATA A6,80           :'72F2  P6     :  LDA  ,X+       ;AUTO INC X
1740 DATA A6,A3           :'72F4            LDA  ,Y--      ;AUTO 2 DEC Y
1750 DATA 35,04           :'72F6            PULS B         ;PULL B (COUNTOR)
1760 DATA 5A              :'72F8            DECB           ;B=B-1
1770 DATA 26,A8           :'72F9            BNE MYCLOOP    ;B>0 THEN JMP MYCLOOP
1780 DATA 39              :'72FB            RTS
1790 DATA 0,0,0,0,0,0,0,0
1800 '
1810 DATA E6,47           :'7304 BITEXTD: LDB 7,U          ;LOAD INPUT PARAMATOR
1820 DATA C5,04           :'7306            BITB #$04      ;DH%=0?
1830 DATA 26,1E           :'7308            BNE QUAD       ;NO THEN JMP QUAD
1840 DATA C6,04           :'730A DOUBLE : LDB #$04         ;SET COUNTOR
1850 DATA A7,44           :'730C L1      :  STA 4,U        ;STORE BASICDATA TO W.A
1860 DATA 8D,38           :'730E            BSR ROTR
1870 DATA 8D,36           :'7310            BSR ROTR
1880 DATA 5A              :'7312            DECB           ;B=B-1
1890 DATA 26,F7           :'7313            BNE L1         ;B<>0 JMP L1
1900 DATA 31,A8,FF        :'7315            LEAY -1,Y      ;Y=Y-1
1910 DATA C6,04           :'7318            LDB #04        ;SET COUNTOR
1920 DATA A7,44           :'731A       L2:  STA 4,U        ;STORE BASICDATA TO W.A
1930 DATA 8D,2A           :'731C            BSR ROTR
1940 DATA 8D,28           :'731E            BSR ROTR
1950 DATA 5A              :'7320            DECB           ;B=B-1
1960 DATA 26,F7           :'7321            BNE L2         ;B<>0 JMP L2
1970 DATA 31,A8,FF        :'7323            LEAY -1,Y
1980 DATA 39              :'7326            RTS
1990 DATA 0
2000 DATA C6,04           :'7328 QUAD    :  LDB #$04       ;SET B=4(COUNTOR)
2010 DATA 34,04           :'732A LQ      :  PSHS B         ;PUSH B
2020 DATA C6,02           :'732C            LDB #$02       ;SET COUNTOR
2030 DATA A7,44           :'732E LQ1     :  STA 4,U        ;LD BASICDT TO W.A
2040 DATA 8D,16           :'7330            BSR ROTR
2050 DATA 8D,14           :'7332            BSR ROTR
2060 DATA 8D,12           :'7334            BSR ROTR
2070 DATA 8D,10           :'7336            BSR ROTR
2080 DATA 5A              :'7338            DECB           ;B=B-1
2090 DATA 26,F3           :'7339            BNE LQ1
2100 DATA 31,A8,FF        :'733B            LEAY -1,Y      ;AUTO DEC Y
2110 DATA 35,04           :'733E            PULS B         ;PULL B
2120 DATA 5A              :'7340            DECB           ;B=B-1
2130 DATA 26,E7           :'7341            BNE LQ
2140 DATA 39              :'7343            RTS
2150 DATA 0,0,0,0
2160 DATA A6,44           :'7348 ROTR    :  LDA 4,U        ;LD DATA
2170 DATA 48              :'734A            ASLA           ;SIFT L WITH CF
2180 DATA 66,A4           :'734B            ROR  ,Y        ;ROTATE DATA MEMORY
2190 DATA 39              :'734D            RTS
2200 '
2210 '
2220 DATA 8E,72,48        :'7240            LDX #RCBMLF    ;=$7248
2230 DATA AD,9F,FB,FA     :'7243            JSR [$FBFA]
2240 DATA 39              :'7247            RTS
2250 DATA 0E              :'7248 RCBMLF : LPOUT CMD
2260 DATA 0               :'7249            DON`T CARE
2270 DATA 72,4E           :'724A            RCBDBA=$724E   ;=$724E
2280 DATA 00,02           :'724C            RCBLNH=2
2290 DATA 1B,41           :'724E RCBMLFDB:ESC,"A"
2300 '********* END OF PROGRAM CCOPY **********
2310 ' SAVE "WCCOPY"

 ******************** ﾁｭｳｲ *************************
 *           ｳｴﾉ LIST ﾁｭｳﾉ   █, █, ＼  ﾊ             *
 *             FM 7,11ﾃﾞﾊ    -, ., ¥  ﾃﾞｽ            *
 **************************************************
```

注：350行　モード␣640

リスト3　CP/M用　CCOPY

```
ICCOPY.COM
-R
NEXT  PC
0480 0100
-D100,40F
0100 21 DB 02 CD 80 02 21 E0 02 CD 80 02 11 00 05 21 !.....!........!
0110 E9 02 06 18 CD 8F 02 11 A0 05 21 01 03 06 16 CD ..........!.....
0120 8F 02 11 20 07 21 17 03 06 86 CD 8F 02 11 B0 07 ... .!..........
0130 21 9D 03 06 10 CD 8F 02 06 04 11 C0 05 C5 D5 21 !..............!
0140 AD 03 06 06 CD 8F 02 D1 21 56 00 19 E5 D1 C1 10 ........!V......
0150 EC 3E 00 32 CE 02 21 00 00 22 D5 02 22 D9 02 21 .>.2..!.."..".."!
0160 B3 03 CD 96 02 CD A5 02 32 D1 02 21 D1 03 CD 96 ........2..!....
0170 02 CD A5 02 32 D2 02 21 F1 03 CD 80 02 3E 00 32 ....2..!.....>.2
0180 CF 02 21 00 00 22 D3 02 21 3F 01 22 D7 02 CD 1B ..!.."..!?."....
0190 02 3E 00 32 D0 02 3A D1 02 87 47 3A D0 02 80 57 .>.2..:...G:...W
01A0 CD 6C 02 21 40 01 22 D3 02 21 7F 02 22 D7 02 CD .l.!@."..!.."...
01B0 1B 02 3E 01 32 D0 02 3A D1 02 87 47 3A D0 02 80 ..>.2..:...G:...
01C0 57 CD 6C 02 3A D2 02 3C F5 06 04 21 C2 05 3A CF W.l.:..<...!..:.
01D0 02 C6 30 77 11 56 00 19 C6 04 10 F7 16 00 CD 62 ..0w.V.........b
01E0 02 3A CF 02 3C 32 CF 02 F1 3D 20 DC 2A D5 02 23 .:..<2...= .*..#
01F0 22 D5 02 22 D9 02 3A CF 02 FE 04 C2 82 01 16 00 "..".:..........
0200 CD 76 02 3A CE 02 3C 32 CE 02 21 F1 03 CD 80 02 .v.:..<2..!.....
0210 3A CE 02 FE 64 C2 7D 01 C3 00 00 DD 21 10 05 DD :...d.}.....!...
0220 36 04 00 DD 36 05 0B DD 21 20 05 DD 36 02 1D 2A 6...6...! ..6..*
0230 D3 02 DD 74 03 DD 75 04 2A D5 02 DD 74 05 DD 75 ...t..u.*...t..u
0240 06 2A D7 02 DD 74 07 DD 75 08 2A D9 02 DD 74 09 .*...t..u.*...t.
0250 DD 75 0A 16 00 C3 58 02 0E 7F 3E FE 21 00 05 C3 .u....X...>.!...
0260 05 00 0E 7F 3E FE 21 A0 05 C3 05 00 0E 7F 3E FE ....>.!.......>.
0270 21 20 07 C3 05 00 0E 7F 3E FE 21 B0 07 C3 05 00 ! ......>.!.....
0280 7E FE 00 C8 E5 5F 0E 05 CD 05 00 E1 23 18 F1 7E ~...._.......#..~
0290 12 13 23 10 FA C9 7E FE 00 C8 E5 5F 0E 02 CD 05 ..#...~...._....
02A0 00 E1 23 18 F1 0E 01 CD 05 00 F5 1E 0D 0E 02 CD ..#.............
02B0 05 00 1E 0A 0E 02 CD 05 00 F1 FE 30 28 0C FE 31 ...........0(..1
02C0 28 08 21 F3 03 CD 96 02 18 DB 06 30 90 C9 55 55 (.!........0..UU
02D0 55 55 55 55 55 55 55 55 55 55 55 1B 57 30 39 00 UUUUUUUUUUU.W09.
02E0 1B 48 30 30 31 31 30 35 00 8E 05 10 AD 9F FB FA .H001105........
02F0 39 55 55 55 55 55 55 55 55 11 00 05 20 00 0B 00 9UUUUUUUU... ...
0300 7C 8E 05 B0 AD 9F FB FA 39 55 55 55 55 55 55 55 |.......9UUUUUUU
0310 55 0E 00 05 C0 01 58 12 12 8E 05 24 33 89 01 F4 U.....X....$3...
0320 31 89 00 A2 AF C4 85 01 27 03 31 A8 28 10 AF 42 1.......'.1.(..B
0330 C6 28 85 02 26 13 A6 84 AA 88 28 AA 88 50 43 A7 .(..&.....(..PC.
0340 A0 A6 80 5A 26 F0 6E CC 0F A6 84 A4 88 28 A4 88 ...Z&.n......(..
0350 50 A7 A0 A6 80 5A 26 F1 AE C4 10 AE 42 C6 28 34 P....Z&.....B.(4
0360 04 A6 84 AA 88 28 E6 88 50 53 E7 44 A4 44 A7 A8 .....(..PS.D.D..
0370 56 A6 88 28 AA 88 50 E6 84 53 E7 45 A4 45 A7 A9 V..(..P..S.E.E..
0380 00 AC A6 84 AA 88 50 E6 88 28 53 E7 46 A4 46 A7 ......P..(S.F.F.
0390 A9 01 02 A6 80 A6 A0 35 04 5A 26 C3 39 8E 07 B8 .......5.Z&.9...
03A0 AD 9F FB FA 39 0E 00 07 BE 00 02 1B 41 1B 49 4D ....9.......A.IM
03B0 30 38 30 4E 4F 4E 20 49 4E 56 45 52 54 20 42 4C 080NON INVERT BL
03C0 41 43 4B 3D 30 2C 49 4E 56 45 52 54 3D 31 20 3F ACK=0,INVERT=1 ?
03D0 00 44 49 53 50 52 41 59 20 4D 4F 44 45 20 36 34 .DISPRAY MODE 64
03E0 30 2A 34 30 30 3D 30 2C 2A 32 30 30 3D 31 20 3F 0*400=0,*200=1 ?
03F0 00 18 00 52 45 54 52 59 21 20 2D 2D 2D 20 50 55 ...RETRY! --- PU
0400 53 48 20 30 20 4F 52 20 31 20 2D 2D 2D 20 00 A7 SH 0 OR 1 --- ..
```

リスト4　CP/M用　WCCOPY

```
IWCCOPY.COM
-R
NEXT  PC
0500 0100
-D100,497
0100 21 EB 02 CD 8F 02 21 F0 02 CD 8F 02 11 00 05 21 !.....!........!
0110 F9 02 06 18 CD 9E 02 11 70 05 21 11 03 06 16 CD ........p.!.....
0120 9E 02 11 50 07 21 27 03 06 FE CD 9E 02 11 40 07 ...P.!'.......@.
0130 21 25 04 06 10 CD 9E 02 06 04 11 90 05 C5 D5 21 !%.............!
0140 35 04 06 06 CD 9E 02 D1 21 6A 00 19 E5 D1 C1 10 5.......!j......
0150 EC 21 00 00 22 DD 02 22 E3 02 22 E7 02 21 3B 04 .!.."..".."..!;.
0160 CD A5 02 CD B4 02 32 E1 02 21 59 04 CD A5 02 CD ......2..!Y.....
0170 B4 02 32 E2 02 21 79 04 CD 8F 02 3E 00 32 DF 02 ..2..!y....>.2..
0180 21 00 00 22 E5 02 21 C7 00 22 E9 02 CD 2A 02 3E !.."..!.."...*.>
0190 00 32 E0 02 3A E2 02 87 87 47 3A E1 02 87 80 47 .2..:....G:....G
01A0 3A E0 02 80 57 CD 7B 02 3A E2 02 FE 01 28 28 21 :...W.{.:....((!
01B0 C8 00 22 E5 02 21 8F 01 22 E9 02 CD 2A 02 3E 01 .."..!.."...*.>.
01C0 32 E0 02 3A E2 02 87 87 47 3A E1 02 87 80 47 3A 2..:....G:....G:
01D0 E0 02 80 57 CD 7B 02 3E 02 F5 06 04 21 92 05 3A ...W.{.>....!..:
01E0 DF 02 C6 30 77 11 6A 00 19 C6 04 10 F7 16 00 CD ...0w.j.........
01F0 71 02 3A DF 02 3C 32 DF 02 F1 3D 20 DC 2A E3 02 q.:..<2...= .*..
0200 23 22 E3 02 22 E7 02 3A DF 02 FE 04 C2 80 01 16 #"..".:.........
0210 00 CD 85 02 2A DD 02 23 22 DD 02 7C FE 01 C2 7B ....*..#"..|...{
0220 01 7D FE 40 C2 7B 01 C3 00 00 DD 21 10 05 DD 36 .}.@.{.....!...6
0230 04 00 DD 36 05 0B DD 21 20 05 DD 36 02 1D 2A E3 ...6...! ..6..*.
0240 02 DD 74 03 DD 75 04 2A E5 02 DD 74 05 DD 75 06 ..t..u.*...t..u.
0250 2A E7 02 DD 74 07 DD 75 08 2A E9 02 DD 74 09 DD *...t..u.*...t..
0260 75 0A 16 00 C3 67 02 0E 7F 3E FE 21 00 05 C3 05 u....g...>.!....
0270 00 0E 7F 3E FE 21 70 05 C3 05 00 0E 7F 3E FE 21 ...>.!p......>.!
0280 50 07 C3 05 00 0E 7F 3E FE 21 40 07 C3 05 00 7E P......>.!@....~
0290 FE 00 C8 E5 5F 0E 05 CD 05 00 E1 23 18 F1 7E 12 ...._......#..~.
```

リスト4　CP/M用　WCCOPY

```
02A0 13 23 10 FA C9 7E FE 00 C8 E5 5F 0E 02 CD 05 00 .#...~...._.....
02B0 E1 23 18 F1 0E 01 CD 05 00 F5 1E 0D 0E 02 CD 05 .#..............
02C0 00 1E 0A 0E 02 CD 05 00 F1 FE 30 28 0C FE 31 28 ..........0(..1(
02D0 08 21 7B 04 CD A5 02 18 DB 06 30 90 C9 32 76 38 .!{.......0..2v8
02E0 C9 CD 5A 19 CD 24 24 CD 49 1E CD 1B 57 30 39 00 ..Z..$$.I...W09.
02F0 1B 48 30 30 31 31 30 35 00 8E 05 10 AD 9F FB FA .H001105........
0300 39 37 3D B8 C2 94 25 32 72 11 00 05 20 00 0B 00 97=...%2r... ...
0310 4F 8E 05 80 AD 9F FB FA 39 C3 F3 0A CD 7A 04 C9 O.......9....z..
0320 3A 0E 00 05 90 01 A8 12 12 8E 05 24 33 89 02 14 :..........$3...
0330 31 89 00 D5 AF C4 85 01 27 03 31 A8 CE 10 AF 42 1.......'.1....B
0340 C6 19 A7 47 85 02 26 18 34 04 A6 84 AA 88 19 AA ...G..&.4.......
0350 88 32 43 17 00 85 A6 80 35 04 5A 26 EB 6E CC 13 .2C.....5.Z&.n..
0360 34 04 A6 84 A4 88 19 A4 88 32 8D 6F A6 80 35 04 4........2.o..5.
0370 5A 26 ED AE C4 10 AE 42 C6 19 34 04 A6 84 AA 88 Z&.....B..4.....
0380 19 E6 88 32 53 E7 44 A4 44 34 20 31 A9 00 6A 8D ...2S.D.D4 1..j.
0390 4A 35 20 A6 88 19 AA 88 32 E6 84 53 E7 45 A4 45 J5 .....2..S.E.E
03A0 34 20 31 A9 00 D4 8D 33 35 20 A6 84 AA 88 32 E6 4 1....35 ....2.
03B0 88 19 53 E7 46 A4 46 34 20 31 A9 01 3E 8D 1C 35 ..S.F.F4 1..>..5
03C0 20 E6 47 C5 04 27 02 A6 A3 A6 80 A6 A3 35 04 5A  .G..'.......5.Z
03D0 26 A8 39 B7 C2 58 25 C1 AF 21 34 E6 47 C5 04 26 &.9..X%..!4.G..&
03E0 1E C6 04 A7 44 8D 38 8D 36 5A 26 F7 31 A8 FF C6 ....D.8.6Z&.1...
03F0 04 A7 44 8D 2A 8D 28 5A 26 F7 31 A8 FF 39 26 C6 ..D.*.(Z&.1..9&.
0400 04 34 04 C6 02 A7 44 8D 16 8D 14 8D 12 8D 10 5A .4....D........Z
0410 26 F3 31 A8 FF 35 04 5A 26 E7 39 E5 2A C1 37 A6 &.1..5.Z&.9.*.7.
0420 44 48 66 A4 39 8E 07 48 AD 9F FB FA 39 0E 00 07 DHf.9..H....9...
0430 4E 00 02 1B 41 1B 49 4D 31 30 30 4E 4F 4E 20 49 N...A.IM100NON I
0440 4E 56 45 52 54 20 42 4C 41 43 4B 3D 30 2C 49 4E NVERT BLACK=0,IN
0450 56 45 52 54 3D 31 20 3F 00 44 49 53 50 52 41 59 VERT=1 ?.DISPRAY
0460 20 4D 4F 44 45 20 36 34 30 2A 34 30 30 3D 30 2C  MODE 640*400=0,
0470 2A 32 30 30 3D 31 20 3F 00 18 00 52 45 54 52 59 *200=1 ?...RETRY
0480 21 20 2D 2D 2D 20 50 55 53 48 20 30 20 4F 52 20 ! --- PUSH 0 OR
0490 31 20 2D 2D 2D 20 00 5A 1 --- .Z
```

RANDOM BOX

FM-7/11将棋対局をより面白く！

■西野直樹

'83年12月号FM-7/8将棋対局に手合割の機能を加えましたので発表させていただきます．

マイコンでの将棋対局は，まだレベルが低く，負けた人は少ないと思います．そこで，現段階でなんとか互角に勝負できないかと思い考えついたのがこのプログラムです．

プログラムは，データのうち$1200～$13CD，メイン・プログラム，サブプログラムをそれぞれ$200ずつ手前にずらしたので変更前のプログラムから$1200～$34FFと$4D00～$59FFをセーブしておいてください．次にメインプログラムをリスト通り変更してコンパイル化し，オブジェクトをセーブしてください．

リセットの後，以下の手順でロードします．

①LOADM "ファイルネーム", −&H200
……………………………$1200～$34FF

②LOADM "ファイルネーム", −&H1D00
……………………………メイン・プログラム

③LOADM ……………………$4D00～$59FF

④SAVEM "SHOGI" ,&H1000,&H59FF, &H3300として完成です．プログラム・エリアとスタート・アドレスが変わっているので注意してください．

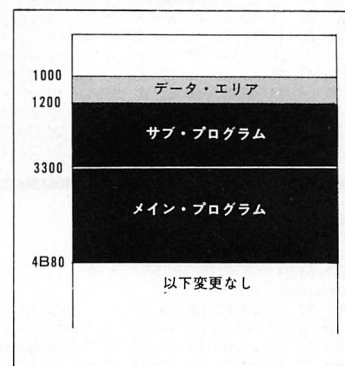
資料には5段差までしかなかったので6段級差からは適当につくりました．

また，駒を落とした側が先手です．

手合割

0段(級)差	平手	7段(級)差	4枚落
1段(級)差	左香落	8段(級)差	5枚落
2段(級)差	角落	9段(級)差	6枚落
3段(級)差	飛落	10段(級)差	7枚落
4段(級)差	飛左香落	11段(級)差	8枚落
5段(級)差	飛角落	12段(級)差	9枚落
6段(級)差	3枚落	13段(級)差	10枚落

🔲参考文献

"将棋手帳" 日本将棋連盟

追加リスト

```
3630 'BN=II+JJ;K=HIGH(PEEK(KDATA);KDATA=KDATA+1;
IF K=10 THEN IF GOTO L3670;FI
3631 'IF K=1 THEN IF TEAI>=3 THEN GOTO L3650;FI;
FI
3632 'IF K=2 THEN IF TEAT=2 OR TEAI>=5 THEN GOTO
 L3650;FI;FI
3633.'IF K=6 THEN IF ((TEAI=1 OR TEAI=4 OR TEAI=
6)AND BN=111)OR TEAI>=7 THEN GOTO L3650;FI;FI
3634''IF K=5 THEN IF(TEAI=8 AND BN=112)OR TEAI>=
9 THEN GOTO L3650;FI;FI
3635 'IF K=4 THEN IF(TEAI=10 AND BN=113)OR TEAI>
=11 THEN GOTO L3650;FI;FI
3636 'IF K=3 THEN IF(TEAI=12 AND BN=114)OR TEAI>
=13 THEN GOTO L3650;FI;FI
3650 'L3650;K=K+15;BN=142-BN;GOSUB L3400
15003 'PRINT " ﾃｱｲ (0-13)";TEAI=INPUT;IF TEAI<=0
 THEN PRINT /," ｾﾝﾃ ... 1",/,/," ｺﾞﾃ ... 2 ";CH=
INPUT;FI
```

変更箇所リスト

```
100 'SUBSYS=$1200
3640 'GOSUB L3400
15002 'PRINT " ｼｮｳｷﾞ ﾀｲｷｮｸ Ver 2.1 by MAX & NISH
INO",/,/
61010 'FOR I=0 TO 53;POKE $6500+I,PEEK($1000+I);
NEXT
61020 'FOR I=0 TO 15;POKE $5B40+I,PEEK($1040+I);
NEXT
61050 'A=I*16+J;POKE $5D46+A,PEEK($1050+A)
61070 'FOR I=0 TO 19;POKE $5DE8+I,PEEK($10F0+I);
NEXT
61110 'POKE $6304+I,PEEK($1110+I)
61180 'FOR I=0 TO 59;POKE $6020+I,PEEK($1160+I);
NEXT
61190 'FOR I=0 TO 45;POKE $6060+I,PEEK($11A0+I);
NEXT
```

予測計算プログラム
回帰計算をBASICで

■JG1INE

そろそろマンネリ化してきたところで，仕事に使えるプログラムを作ろうと思い，需要予測するプログラムを作りました．

人口，事務量，物理実験などの実験観測値を時系列的に分析し，4種類の方程式にあてはめることにより，将来の値を予測するプログラムです．

プログラムの説明

プログラムはすべてBASICで記述してあり，他の機種への移植も容易にできます．FMは，算術計算の精度が小数点以下6桁目以降は精度が悪くなるようですが，実用上は問題はないと思います（FMの関数の演算は，単精度で計算しているため．ただし，プログラム中の変数はすべて倍精度です）．

プログラムはメニュー方式（画面で指示）としており，漢字ROMを実装しているときは画面のみ漢字表示し，実装していないときはカナで表示します．ただし，プリンタへの印字は，漢字ROMを実装していても漢字を使うと，印字速度が極端に遅くなり実用にならないため，カナ文字で印字することにしました．

予測できる方程式は，直線，指数曲線，対数曲線およびべき乗曲線の4種類です．予測する式を選択した後，データ入力指示終了で方程式を求め，その後の変数xの入力で予測結果を出力します．

本プログラムでは，得られた方程式が入力データにどの程度一致しているか，わかるように決定指数（R2で表示）も計算し，予測式のあてはめが適合しているかどうか判断の参考になるようにしています．

プログラム・リストの説明を表1，2に示すので，使いにくいと思う人は改造して，使ってください．

操作方法

プログラムをRUNさせると，プリンタに出力するか聞いてくるので，"Y" または "N" で答えてください．つぎに，漢字ROMを実装しているか聞いてくるので，これにも"Y" または "N" を入力します．

これで，準備OKです．メニュー画面（図1）が表示されるので，予測する式の番号を入力してください．あとは，

図1 （漢字ROM実装時）のタイトル画像

```
******** 予 測 計 算 プ ロ グ ラ ム *******

予測計算する曲線の番号を入力して下さい

1・・・直　線　　　　　　2・・・指数曲線

3・・・対数曲線　　　　　4・・・べき乗曲線

5・・・終　了
```

画面の指示に従い予測するデータを入力し，データがなくなったらxの値として "9999" を入力します．"9999" を入力すると，方程式を計算して画面に表示（プリンタに出力指定したときは，印刷します）し，予測データの入力待ちになります．

予測データの入力（xの値）をすると予測値を計算し，計算結果をCRTに表示（プリンタに出力指定したときは，xの値と予測結果を印字します）します．予測データがなくなったときは，xの値に "9999" を入力すると，メニュー画面を表示し，次の予測データの入力が可能になります．

実行したときのプリント・アウトを図2に示すので，自分でプログラム・リストを入力したときは，テスト・データとして結果を比較すれば，入力したプログラムが正常かどうかわかると思います．

終わりに

以上で本プログラムの説明は終わりますが，xおよびyに "0" を入力すると，"Division By Zero" のエラーになるので，入力データに "0" があるときは，xの値を0以外に変換(例：－2，－1，0，1，2，3→1，2，3，4，5，6）した後，データを入力してください．

最後に，このプログラムが少しでも，仕事の役にたてば幸いです．

参考文献

1）F-BASIC文法書
2）YHPモデル25P　プログラムライブラリー
3）"コンピュータマネジメントサイエンスハンドブック"：（オーム社）

表1　変数の説明

変数	説明
K$	キー入力変数
PR	0：プリンタなし　1：プリンタあり
KR	0：漢字ROMなし　1：漢字ROMあり
A1#	一次式の定数
A0#	一次式の定数
R1#	一次式の決定係数
X#	予測データ
Y#	予測データ
A2#	指数曲線の定数
A3#	指数曲線の定数
R2#	指数曲線の決定定数
A4#	対数曲線の定数
A5#	対数曲線の定数
R3#	対数曲線の決定定数
A6#	べき乗曲線の定数
A7#	べき乗曲線の定数
R4#	べき乗曲線の決定定数
A#	X#の累積値
B#	X#の2乗の累積値
C#	Y#の累積値
D#	X#×Y#の累積値
E#	Y#の2乗の累積値
ALY#	X#×log(Y#)の累積値
LY#	log(Y#)の累積値
LY2#	log(Y#)の2乗の累積値
CLX#	Y#×log(X#)の累積値
LX#	log(X#)の累積値
LX2#	log(X#)の2乗の累積値
LXLY#	log(X#)×log(Y#)の累積値
N#	データの数

表2　リストの説明

行番号		説明
10～ 90		REM文
100～ 120		プリンタの有無の処理
130～ 150		漢字の有無の処理
160～ 170		漢字ROMを実装していないときの，式選択メニュー画面
180～ 240		式選択後の処理選択ルーチン
250～ 280		漢字ROMを実装しているときの式選択メニュー画面
290～ 580		一次式の処理ルーチン
	290～ 310	漢字ROM実装，プリンタの有無の判断ルーチン
	320～ 330	画面表示ルーチン
	340	サブルーチン(1500～1730…･データ入力ルーチン，予測式へのあてはめおよび決定係数の計算)にジャンプ
	350～ 400	予測式の画面出力およびプリンタに出力
	410～ 560	予測データの入力，予測の計算および出力
	570～ 580	初期画面に
590～ 880		指数曲線の処理ルーチン
	590～ 610	漢字ROM実装，プリンタの有無の処理ルーチン
	620～ 630	画面表示ルーチン
	640	サブルーチン（1500～1730……データ入力ルーチン，予測式へのあてはめおよび決定係数の計算にジャンプ)
	650～ 700	予測式の画面出力およびプリンタ出力
	710～ 860	予測データの入力，予測の計算および出力
	870～ 880	初期画面に
890～1180		対数曲線の処理ルーチン
	890～ 910	漢字ROM実装，プリンタの有無の判断ルーチン
	920～ 930	画面表示ルーチン
	940	サブルーチン（1500～1730……データ入力ルーチン，予測式へのあてはめおよび決定係数の計算）へジャンプ
	950～1000	予測式の画面への出力およびプリンタ出力
	1010～1160	予測データの入力，予測の計算および出力
	1170～1180	初期画面に
1190～1480		べき乗曲線の処理ルーチン
	1190～1210	漢字ROM実装，プリンタの有無の判断ルーチン
	1220～1230	画面表示ルーチン
	1240	サブルーチン（1500～1730……データ入力ルーチン，予測式への当てはめおよび決定係数の計算）にジャンプ
	1250～1300	予測式の画面出力およびプリンタ出力
	1310～1460	予測データの入力，予測の計算および出力
	1470～1480	初期画面へ
1490		終了処理
1500～1730		データ入力ルーチン，予測式への当てはめおよび決定係数の計算サブルーチン
	1500～1550	漢字ROMの有無，プリンタの有無の判断ルーチン
	1560～1610	プリンタがあるときのデータ入力ルーチン
	1620～1630	各種データの累積値の計算ルーチン
	1640～1670	プリンタがないときのデータ入力ルーチン
	1680～1690	各種データの累積値の計算ルーチン
	1700	一次式の計算
	1710	指数曲線の計算
	1720	対数曲線の計算
	1730	べき乗曲線の計算

図2　予測計算例

```
********** チョクセンカイキ **********

        X                Y

    40.500000       104.500000
    38.600000       102.000000
    37.900000       100.000000
    36.200000        97.500000
    35.100000        95.500000
    34.600000        94.000000
  9999.000000
------------------------------------------------------

Y = 1.760312334898099 * X + ( 33.52106342520229 )

R^2= .9906662692890731

---------- ヨ ソ ク チ ----------

        X                Y
    45.000000       112.735118
    50.000000       121.536680
    55.000000       130.338242
    60.000000       139.139804
    70.000000       156.742927


*********** シスウカイキ ***********
```

```
        X                Y

     0.720000         2.160000
     1.310000         1.610000
     1.950000         1.160000
     2.580000         0.850000
     3.140000         0.500000
  9999.000000
------------------------------------------------------

Y = 3.445081472396851 * e ^ (-.5820248764581503 * X)

R^2= .9803259227322417

---------- ヨ ソ ク チ ----------

        X                Y
     4.500000         0.251035
     5.000000         0.187650
     5.500000         0.140270
     6.000000         0.104852


*********** タイスウカイキ ***********

        X                Y

     3.000000         1.500000
     4.000000         9.300000
     6.000000        23.400000
```

図2　予測計算例

```
          10.000000                   45.800000
          12.000000                   60.100000
        9999.000000
--------------------------------------------------------

Y =-47.02073118751196 + ( 41.39420920761625 ) * LOG X

R^2= .9798279748480709

---------- ヨ ソ ク チ ----------

              X                          Y
          13.000000                   59.153326
          14.000000                   62.220963
          15.000000                   65.076877
          16.000000                   67.748387
          17.000000                   70.257894

******** ヘ゛キシ゛ョウカイキ *******

              X                          Y
```

```
          10.000000                   0.950000
          12.000000                   1.050000
          15.000000                   1.250000
          17.000000                   1.410000
          20.000000                   1.730000
          22.000000                   2.000000
          25.000000                   2.530000
          27.000000                   2.980000
          30.000000                   3.850000
          32.000000                   4.590000
          35.000000                   6.020000
        9999.000000
--------------------------------------------------------

Y = 2.622252888977528D-02 * X ^ ( 1.455517479440697 )

R^2= .935492665331433

---------- ヨ ソ ク チ ----------

              X                          Y
          36.000000                   4.829478
          37.000000                   5.025966
          38.000000                   5.224894
          39.000000                   5.426216
```

予測計算プログラム

```
10 '*******************************
20 '*                             *
30 '*   ヨソク ケイサン フ゜ロク゛ラム (FM-8)   *
40 '*                             *
50 '*          1983/10/10         *
60 '*                             *
70 '*                 by JG1INE   *
80 '*                             *
90 '*******************************
100 CLS:WIDTH80:COLOR 4:PRINT "フ゜リンター ニ シュツリョク シ
マスカ ? (Y or N)"
110 K$=INKEY$:IF K$="" THEN 110
120 IF K$="Y" OR K$="y" THEN OPEN"O",#1,"LPT0:":
PR=1
130 PRINT "カンシ゛ ROM ヲ シ゛ッソウ シテイマスカ ? (Y or N)"
140 K$=INKEY$:IF K$="" THEN 140
150 IF K$="Y" OR K$="y" THEN KR=1:GOTO 250
160 CLS:SYMBOL(16,10),"***** ヨ ソ ク ケ イ サ ン フ゜ ロ
ク゛ ラ ム *****",2,2,4:SYMBOL(16,34),"ヨソクケイサン スル シキ
 ノ ハ゛ンコ゛ウ ヲ ニュウリョク シテクタ゛サイ",1.5,1.7:SYMBOL(48,72
),"1・・・チョクセン",2,2:SYMBOL(320,72),"2・・・シスウキョクセン",
2,2
170 SYMBOL(48,112),"3・・・タイスウキョクセン",2,2:SYMBOL(32
0,112),"4・・・ヘ゛キシ゛ョウキョクセン",2,2:SYMBOL(48,152),"5・
・・シュウリョウ",2,2
180 K$=INKEY$:IF K$="" THEN 180
190 IF K$="1" THEN 290
200 IF K$="2" THEN 590
210 IF K$="3" THEN 890
220 IF K$="4" THEN 1190
230 IF K$="5" THEN 1490
240 GOTO 180
250 CLS:PRINT@(32,10),&H2176,&H2176,&H2176,&H217
6,&H2176,&H2176,&H2176,&H2176,&H2120,&H4D3D,&H21
20,&H422C,&H2120,&H3757,&H2120,&H3B3B,&H2120,&H2
557,&H2120,&H256D,&H2120,&H2530,&H2120,&H2569,&H
2120,&H2560,&H2120,&H2176,&H2176,&H2176,&H2176,&
H2176
260 PRINT@(528,10),&H2176,&H2176,&H2176:PRINT@(4
8,44),&H4D3D,&H422C,&H3757,&H3B3B,&H2439,&H246B,
&H364A,&H407E,&H244E,&H4856,&H3966,&H2472,&H467E
,&H4E4F,&H2437,&H2446,&H323C,&H2435,&H2424
270 PRINT@(48,72),&H2331,&H2126,&H2126,&H2126,&H
443E,&H2120,&H2120,&H407E:PRINT@(320,72),&H2332,
&H2126,&H2126,&H2126,&H3B58,&H3F74,&H364A,&H407E
:PRINT@(48,112),&H2333,&H2126,&H2126,&H2126,&H42
50,&H3F74,&H364A,&H407E
280 PRINT@(320,112),&H2334,&H2126,&H2126,&H2126,
&H2459,&H242D,&H3E68,&H364A,&H407E:PRINT@(48,152
),&H2335,&H2126,&H2126,&H2126,&H3D2A,&H2120,&H21
20,&H4E3B:GO TO 180
290 CLS:IF PR=1ANDKR=1 THEN PRINT #1:PRINT #1:PR
INT #1:PRINT #1,CHR$(&HE);"********* チョクセンカイキ **
*******":GOTO 330
300 IF PR=1ANDKR=0 THEN PRINT #1:PRINT #1:PRINT
#1:PRINT #1,CHR$(&HE);"********* チョクセンカイキ ******
***":SYMBOL(16,10),"******** チョクセンカイキ ********",
2,2:GO TO 340
310 IF PR=0ANDKR=1 THEN GO TO 330
320 SYMBOL(16,10),"******** チョクセンカイキ ********",2
,2:GO TO 340
330 CLS:PRINT@(48,16),&H2176,&H2176,&H2176,&H217
6,&H2176,&H2176,&H2176,&H2176,&H2120,&H443E,&H21
20,&H407E,&H2120,&H3273,&H2120,&H3522,&H2120,&H2
176,&H2176,&H2176,&H2176,&H2176,&H2176,&H2176,&H
2176
340 GOSUB 1500
350 PRINT:PRINT:PRINT"Y =";:PRINT A1#;:PRINT"* X
 + (";:PRINT A0#;:PRINT")"
360 PRINT:PRINT:PRINT"R^2=";:PRINT R1#:PRINT:PRI
NT
370 IF PR=0 THEN 500
380 PRINT #1:PRINT #1,"-------------------------
------------------------------"
390 PRINT #1:PRINT #1,"Y =";:PRINT #1,A1#;:PRINT
 #1,"* X + (";:PRINT #1,A0#;:PRINT #1,")"
400 PRINT #1:PRINT #1,"R^2=";:PRINT #1,R1#
410 PRINT #1:PRINT #1:PRINT #1,CHR$(&HE);"------
---- ヨ ソ ク チ ---------":PRINT #1:PRINT #1,TAB(15
)"X";:PRINT #1,TAB(35)"Y"
420 IF KR=1 THEN 440
430 PRINT:PRINT:PRINT:PRINT"ヨソクスル テ゛ータ ヲ ニュウリョク
シテクタ゛サイ (END:X=9999)":PRINT:PRINT:GO TO 450
440 PRINT:PRINT:PRINT:PRINT:PRINT@(32,168),&H4D3
D,&H422C,&H2439,&H246B,&H2547,&H215D,&H253F,&H24
72,&H467E,&H4E4F,&H2437,&H2446,&H323C,&H2435,&H2
424:PRINT:PRINT"(END:X=9999)":PRINT
450 INPUT "X        ";X#
460 IF X#=9999 THEN 570
470 PRINT #1,TAB(7);:PRINT #1,USING "######.####
##";X#;
480 Y#=A1#*X#+A0#:PRINT TAB(20) "Y =";Y#:PRINT #
1,TAB(27);:PRINT #1,USING "######.######";Y#
490 GOTO 450
500 IF KR=1 THEN 520
510 PRINT:PRINT:PRINT:PRINT"ヨソクスル テ゛ータ ヲ ニュウリョク
シテクタ゛サイ (END:X=9999)":PRINT:PRINT:GO TO 530
520 PRINT:PRINT:PRINT:PRINT:PRINT@(32,168),&H4D3
D,&H422C,&H2439,&H246B,&H2547,&H215D,&H253F,&H24
72,&H467E,&H4E4F,&H2437,&H2446,&H323C,&H2435,&H2
424:PRINT:PRINT"(END:X=9999)":PRINT
530 INPUT "X        ";X#
540 IF X#=9999 THEN 570
550 Y#=A1#*X#+A0#:PRINT TAB(20) "Y =";Y#
560 GOTO 530
570 IF KR=1 THEN 250
580 GOTO 160
590 CLS:IF PR=1ANDKR=1 THEN PRINT #1:PRINT #1:PR
INT #1:PRINT #1,CHR$(&HE);"********** シスウカイキ ***
*******":GOTO 630
600 IF PR=1ANDKR=0 THEN PRINT #1:PRINT #1:PRINT#
1:PRINT #1,CHR$(&HE);"********** シスウカイキ ********
**":SYMBOL(16,10),"********* シスウカイキ *********",2
,2:GO TO 640
610 IF PR=0ANDKR=1 THEN GO TO 630
620 SYMBOL(16,10),"********* シスウカイキ *********",2
,2:GO TO 640
630 CLS:PRINT@(48,16),&H2176,&H2176,&H2176,&H217
6,&H2176,&H2176,&H2176,&H2176,&H2120,&H3B58,&H21
20,&H3F74,&H2120,&H3273,&H2120,&H3522,&H2120,&H2
176,&H2176,&H2176,&H2176,&H2176,&H2176,&H2176,&H
2176
640 GOSUB 1500
650 PRINT:PRINT:PRINT"Y =";:PRINT A2#;:PRINT"* e
 ^ (";:PRINT A3#;:PRINT"* X)"
660 PRINT:PRINT:PRINT"R^2=";:PRINT R2#:PRINT:PRI
NT
670 IF PR=0 THEN 800
680 PRINT #1:PRINT #1,"-------------------------
------------------------------"
690 PRINT #1:PRINT #1,"Y =";:PRINT #1,A2#;:PRINT
 #1,"* e ^ (";:PRINT #1,A3#;:PRINT #1,"* X)"
700 PRINT #1:PRINT #1,"R^2=";:PRINT #1,R2#
710 PRINT #1:PRINT #1:PRINT #1,CHR$(&HE);"------
--- ヨ ソ ク チ ----------":PRINT #1:PRINT #1,TAB(15
)"X";:PRINT #1,TAB(35)"Y"
720 IF KR=1 THEN 740
730 PRINT:PRINT:PRINT:PRINT"ヨソクスル テ゛ータ ヲ ニュウリョク
```

予測計算プログラム　リスト

```
ｼﾃｸﾀﾞｻｲ (END:X=9999)":PRINT:PRINT:GO TO 750
740 PRINT:PRINT:PRINT:PRINT:PRINT@(32,168),&H4D3
D,&H422C,&H2439,&H246B,&H2547,&H215D,&H253F,&H24
72,&H467E,&H4E4F,&H2437,&H2446,&H323C,&H2435,&H2
424:PRINT:PRINT"(END:X=9999)":PRINT
750 INPUT "X        ";X#
760 IF X#=9999 THEN 870
770 PRINT #1,TAB(7);:PRINT #1,USING "######.####
##";X#;
780 Y#=A2#*(EXP(A3#*X#)):PRINT TAB(20) "Y =";Y#:
PRINT #1,TAB(27);:PRINT #1,USING "######.######"
;Y#
790 GOTO 750
800 IF KR=1 THEN 820
810 PRINT:PRINT:PRINT:PRINT"ﾖｿｸｽﾙ ﾃﾞｰﾀ ｦ ﾆｭｳﾘｮｸ
ｼﾃｸﾀﾞｻｲ (END:X=9999)":PRINT:PRINT:GO TO 830
820 PRINT:PRINT:PRINT:PRINT:PRINT@(32,168),&H4D3
D,&H422C,&H2439,&H246B,&H2547,&H215D,&H253F,&H24
72,&H467E,&H4E4F,&H2437,&H2446,&H323C,&H2435,&H2
424:PRINT:PRINT"(END:X=9999)":PRINT
830 INPUT "X        ";X#
840 IF X#=9999 THEN 870
850 Y#=A2#*(EXP(A3#*X#)):PRINT TAB(20) "Y =";Y#
860 GOTO 830
870 IF KR=1 THEN 250
880 GOTO 160
890 CLS:IF PR=1ANDKR=1 THEN PRINT #1:PRINT #1:PR
INT #1:PRINT #1,CHR$(&HE);"********** ﾀｲｽｳｶｲｷ **
********":GOTO 930
900 IF PR=1ANDKR=0 THEN PRINT #1:PRINT #1:PRINT
#1:PRINT #1,CHR$(&HE);"********** ﾀｲｽｳｶｲｷ ******
****":SYMBOL(16,10),"********* ﾀｲｽｳｶｲｷ *********
",2,2:GO TO 940
910 IF PR=0ANDKR=1 THEN GO TO 930
920 SYMBOL(16,10),"********* ﾀｲｽｳｶｲｷ *********",
2,2:GO TO 940
930 CLS:PRINT@(48,16),&H2176,&H2176,&H2176,&H217
6,&H2176,&H2176,&H2176,&H2176,&H2120,&H4250,&H21
20,&H3F74,&H2120,&H3273,&H2120,&H3522,&H2120,&H2
176,&H2176,&H2176,&H2176,&H2176,&H2176,&H2176,&H
2176
940 GOSUB 1500
950 PRINT:PRINT:PRINT"Y =";:PRINT A4#;:PRINT"+ (
";:PRINT A5#;:PRINT") * LOG X"
960 PRINT:PRINT:PRINT"R^2=";:PRINT R3#:PRINT:PRI
NT
970 IF PR=0 THEN 1100
980 PRINT #1:PRINT #1,"-------------------------
-------------------------------"
990 PRINT #1:PRINT #1,"Y =";:PRINT #1,A4#;:PRINT
 #1,"+ (";:PRINT #1,A5#;:PRINT #1,") * LOG X"
1000 PRINT #1:PRINT #1,"R^2=";:PRINT #1,R3#
1010 PRINT #1:PRINT #1:PRINT #1,CHR$(&HE);"-----
---- ﾖ ｿ ｸ ﾁ ----------":PRINT #1:PRINT #1,TAB(1
5)"X";:PRINT #1,TAB(35)"Y"
1020 IF KR=1 THEN 1040
1030 PRINT:PRINT:PRINT:PRINT"ﾖｿｸｽﾙ ﾃﾞｰﾀ ｦ ﾆｭｳﾘｮｸ
 ｼﾃｸﾀﾞｻｲ (END:X=9999)":PRINT:PRINT:GO TO 1050
1040 PRINT:PRINT:PRINT:PRINT:PRINT@(32,168),&H4D
3D,&H422C,&H2439,&H246B,&H2547,&H215D,&H253F,&H2
472,&H467E,&H4E4F,&H2437,&H2446,&H323C,&H2435,&H
2424:PRINT:PRINT"(END:X=9999)":PRINT
1050 INPUT "X        ";X#
1060 IF X#=9999 THEN 1170
1070 PRINT #1,TAB(7);:PRINT #1,USING "######.###
###";X#;
1080 Y#=A4#+A5#*LOG(X#):PRINT TAB(20) "Y =";Y#:P
RINT #1,TAB(27);:PRINT #1,USING "######.######";
Y#
1090 GOTO 1050
1100 IF KR=1 THEN 1120
1110 PRINT:PRINT:PRINT:PRINT"ﾖｿｸｽﾙ ﾃﾞｰﾀ ｦ ﾆｭｳﾘｮｸ
 ｼﾃｸﾀﾞｻｲ (END:X=9999)":PRINT:PRINT:GO TO 1130
1120 PRINT:PRINT:PRINT:PRINT:PRINT@(32,168),&H4D
3D,&H422C,&H2439,&H246B,&H2547,&H215D,&H253F,&H2
472,&H467E,&H4E4F,&H2437,&H2446,&H323C,&H2435,&H
2424:PRINT:PRINT"(END:X=9999)":PRINT
1130 INPUT "X        ";X#
1140 IF X#=9999 THEN 1170
1150 Y#=A4#+A5#*LOG(X#):PRINT TAB(20) "Y =";Y#
1160 GOTO 1130
1170 IF KR=1 THEN 250
1180 GOTO 160
1190 CLS:IF PR=1ANDKR=1 THEN PRINT #1:PRINT #1:P
RINT #1:PRINT #1,CHR$(&HE);"******** ﾍﾞｷｼﾞｮｳｶｲｷ
*******":GOTO 1230
1200 IF PR=1ANDKR=0 THEN PRINT #1:PRINT #1:PRINT
 #1:PRINT #1,CHR$(&HE);"********* ﾍﾞｷｼﾞｮｳｶｲｷ ***
******":SYMBOL(16,10),"******** ﾍﾞｷｼﾞｮｳｶｲｷ *****
***",2,2:GO TO 1240
1210 IF PR=0ANDKR=1 THEN GO TO 1230
1220 SYMBOL(16,10),"******** ﾍﾞｷｼﾞｮｳｶｲｷ ********
",2,2:GO TO 1240
1230 CLS:PRINT@(48,16),&H2176,&H2176,&H2176,&H21
76,&H2176,&H2176,&H2176,&H2120,&H2459,&H2120,&H2
42D,&H2120,&H3E68,&H2120,&H3273,&H2120,&H3522,&H
2120,&H2176,&H2176,&H2176,&H2176,&H2176,&H2176,&
H2176
1240 GOSUB 1500
1250 PRINT:PRINT:PRINT"Y =";:PRINT A6#;:PRINT"*
X ^ (";:PRINT A7#;:PRINT")"
1260 PRINT:PRINT:PRINT"R^2=";:PRINT R4#:PRINT:PR
INT
1270 IF PR=0 THEN 1400
1280 PRINT #1:PRINT #1,"------------------------
-------------------------------"
1290 PRINT #1:PRINT #1,"Y =";:PRINT #1,A6#;:PRIN
T #1,"* X ^ (";:PRINT #1,A7#;:PRINT #1,")"
1300 PRINT #1:PRINT #1,"R^2=";:PRINT #1,R4#
1310 PRINT #1:PRINT #1:PRINT #1,CHR$(&HE);"-----
---- ﾖ ｿ ｸ ﾁ ---------":PRINT #1:PRINT #1,TAB(15
)"X";:PRINT #1,TAB(35)"Y"
1320 IF KR=1 THEN 1340
1330 PRINT:PRINT:PRINT:PRINT"ﾖｿｸｽﾙ ﾃﾞｰﾀ ｦ ﾆｭｳﾘｮｸ
 ｼﾃｸﾀﾞｻｲ (END:X=9999)":PRINT:PRINT:GO TO 1350
1340 PRINT:PRINT:PRINT:PRINT:PRINT@(32,168),&H4D
3D,&H422C,&H2439,&H246B,&H2547,&H215D,&H253F,&H2
472,&H467E,&H4E4F,&H2437,&H2446,&H323C,&H2435,&H
2424:PRINT:PRINT"(END:X=9999)":PRINT
1350 INPUT "X        ";X#
1360 IF X#=9999 THEN 1470
1370 PRINT #1,TAB(7);:PRINT #1,USING "######.###
###";X#;
1380 Y#=A6#*(X#^A7#):PRINT TAB(20) "Y =";Y#:PRIN
T #1,TAB(27);:PRINT #1,USING "######.######";Y#
1390 GOTO 1350
1400 IF KR=1 THEN 1420
1410 PRINT:PRINT:PRINT:PRINT"ﾖｿｸｽﾙ ﾃﾞｰﾀ ｦ ﾆｭｳﾘｮｸ
 ｼﾃｸﾀﾞｻｲ (END:X=9999)":PRINT:PRINT:GO TO 1430
1420 PRINT:PRINT:PRINT:PRINT:PRINT@(32,168),&H4D
3D,&H422C,&H2439,&H246B,&H2547,&H215D,&H253F,&H2
472,&H467E,&H4E4F,&H2437,&H2446,&H323C,&H2435,&H
2424:PRINT:PRINT"(END:X=9999)":PRINT
1430 INPUT "X        ";X#
1440 IF X#=9999 THEN 1470
1450 Y#=A6#*(X#^A7#):PRINT TAB(20) "Y =";Y#
1460 GOTO 1430
1470 IF KR=1 THEN 250
1480 GOTO 160
1490 CLOSE #1:CLS:END
1500 IF KR=1 THEN 1520
1510 SYMBOL(32,40),"ﾃﾞｰﾀ ｦ ﾆｭｳﾘｮｸ ｼﾃｸﾀﾞｻｲ",2,2:L
OCATE 4,6:PRINT"(END:X=9999)":GOTO 1530
1520 PRINT@(32,40),&H2547,&H215D,&H253F,&H2472,&
H467E,&H4E4F,&H2437,&H2446,&H323C,&H2435,&H2424:
LOCATE 4,6:PRINT"(END:X=9999)"
1530 A#=0:B#=0:C#=0:D#=0:E#=0:X#=0:Y#=0:ALY#=0:L
Y#=0:LY2#=0:CLX#=0:LX#=0:LX2#=0:LXLY#=0:N#=0
1540 IF PR=0 THEN 1640
1550 IF X#=9999 THEN 1700
1560 PRINT #1:PRINT #1:PRINT #1,TAB(15)"X";:PRIN
T #1,TAB(35)"Y":PRINT #1:LOCATE 0,8
1570 BEEP:INPUT "X    ";X#
1580 PRINT #1,TAB(7);:PRINT #1,USING "######.###
###";X#;
1590 IF X#=9999 THEN 1700
1600 BEEP:INPUT "Y    ";Y#
1610 PRINT #1,TAB(27);:PRINT #1,USING "######.##
####";Y#
1620 A#=A#+X#:B#=B#+X#^2:C#=C#+Y#:D#=D#+X#*Y#:E#
=E#+Y#^2:ALY#=ALY#+X#*LOG(Y#):LY#=LY#+LOG(Y#):LY
2#=LY2#+(LOG(Y#))^2:CLX#=CLX#+Y#*LOG(X#):LX#=LX#
+LOG(X#):LX2#=LX2#+(LOG(X#))^2:LXLY#=LXLY#+(LOG(
X#))*(LOG(Y#)):N#=N#+1
1630 GO TO 1570
1640 LOCATE 0,8
1650 BEEP:INPUT "X    ";X#
1660 IF X#=9999 THEN 1700
1670 BEEP:INPUT "Y    ";Y#
1680 A#=A#+X#:B#=B#+X#^2:C#=C#+Y#:D#=D#+X#*Y#:E#
=E#+Y#^2:ALY#=ALY#+X#*LOG(Y#):LY#=LY#+LOG(Y#):LY
2#=LY2#+(LOG(Y#))^2:CLX#=CLX#+Y#*LOG(X#):LX#=LX#
+LOG(X#):LX2#=LX2#+(LOG(X#))^2:LXLY#=LXLY#+(LOG(
X#))*(LOG(Y#)):N#=N#+1
1690 GO TO 1650
1700 A1#=(D#-(A#*C#/N#))/(B#-(A#^2/N#)):A0#=C#/N
#-A1#*(A#/N#):R1#=(D#-(A#*C#/N#))^2/((B#-(A#^2/N
#))*(E#-(C#^2/N#)))
1710 A3#=(ALY#-(A#*LY#)/N#)/(B#-(A#^2)/N#):A2#=E
XP(LY#/N#-(A3#*A#)/N#):R2#=((ALY#-(A#*LY#)/N#)^2
)/((B#-(A#^2)/N#)*(LY2#-(LY#^2)/N#))
1720 A5#=(CLX#-(LX#*C#)/N#)/(LX2#-(LX#^2)/N#):A4
#=(C#-A5#*LX#)/N#:R3#=((CLX#-(LX#*C#)/N#)^2)/((L
X2#-(LX#^2)/N#)*(E#-(C#^2)/N#))
1730 A7#=(LXLY#-(LX#*LY#)/N#)/(LX2#-(LX#^2)/N#):
A6#=EXP((LY#/N#)-(A7#*LX#)/N#):R4#=((LXLY#-(LX#*
LY#)/N#)^2)/((LX2#-(LX#^2)/N#)*(LY2#-(LY#^2)/N#)
):RETURN
```

FM-7

ALL RAM DISK BASIC 作成プログラム

FM-7でFM-8のDISK BASICを動かす

■コンピ

FM-8用のDOSモードDISK BASICが発表されていましたが，改良を加え，FM-7/8両用のDOSモードDISK BASICを作るプログラムを作りました．

特徴

❶ALL RAMモードであるので，BASICの拡張，変更が容易．

❷FM-7のインタープリンタをFM-8で走らせることが可能．この逆も成立．つまり，サブCPUに保存しなければ，マシン語のコンパチビリティーを保つことができる．ただし，ハードの都合により，何もない機能はできないが，SYNTAX ERRORなどはでない．

❸FILESをとったときに，最初のファイル名にDOS-7(DOS-8)とカラーで表示されるので，どちらのインタープリタが入っているか一目瞭然．

実行手順

DOSモードDISK作成

1)ブランク・ディスケットに，FM-7(FM-8)のシステム・ディスクの"SYSDSK"というプログラムで，FORMATをかけ，BASICコピーを行なってください．

2)本プログラムをロードし，**EXEC &H6000**⏎で実行します．

3)さきほどのディスケットを，ドライブ0に入れて⏎キーを押します．すると，ALL RAM(DOSモード)DISK-BASICのディスケットができあがります(なお，自動的に**DSKINI0**のコマンドも実行します)．

BASICの拡張・変更

1)DOSモード・ディスクでブートして，何らかの方法で，インタープリタを変更してください．

2)本プログラムをロードし，**EXEC&H6000**⏎を実行します．

3)DOSモード・ディスクをドライブ0に入れ⏎キーを押します．すると，拡張，変更されたBASICのディスケットができあがります．

プログラム

リロケートの必要はないと思い，ポジション固定のプログラムです．必ず6000(H)番地から入力してください．

まず初めに，DISK-BASICの転用部分(FM-7は$6E00～，FM-8は$7000～が，DISK-BASICを起動してしまうと，内部が書き変わってしまいます．そこで，トラック0の15～32セクタをトラック2の1セクタかわのセクタにコピーします．

次に，BASICインタープリタを$8000～$FBFFまですっぽり続きのセクタにコピーします．そのあと，FATの初期化，ディレクトリの初期化を行ない，最後に，IPLを設定して終了します．なお，BASICインタープリタは，トラック2からセーブされますが(通常ここは，ユーザーエリア)，FATにシステム領域と設定しているので，いままで通りに，何ら支障なく使えます．

また，**DSKINI0**を実行しても，"134 Clusters Free"のままなので，**DSKINI0**を実行してもDOSモードBASICは，そのまま使えます(FM-8の場合は，DISK-BASICの都合上，"152 Clusters Free"となります．つまり，FATのシステム領域の設定が破壊されるため，**DSKINI0**実行後はもう一度，初めからDOSモード・ディスクの作成をしてください)．

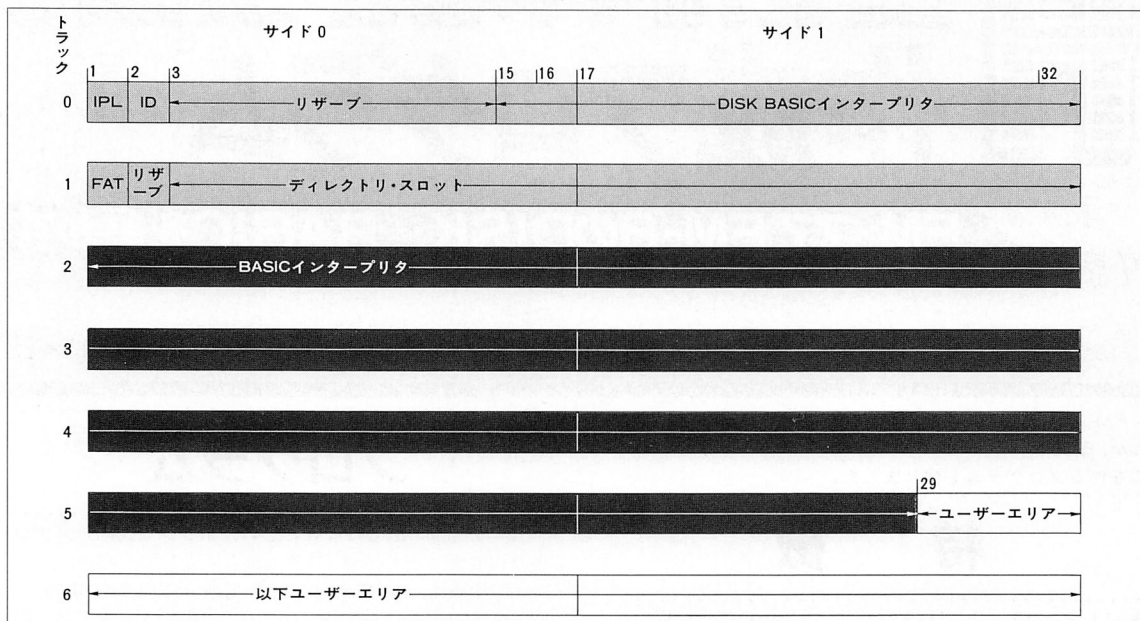
本プログラム実行後のディスケットの構造スロットを図1に示します．

使い方

DISK-BASICではあまり使われない，**MOTOR**の省略形**M.**を**FILES**の**F.**に変えてしまっても大変便利ではないでしょうか．

FM-7	$C724	$4D → $46
	$C726	$A8 → $B0
FM-8	$CA5A	$4D → $46
	$CA5C	$A8 → $B0

また，FM-8用に開発された言語，アプリケーション・ソフトも倍速でFM-7で走るので，大変便利だと思います．

図1　構造スロット

最後に

発売当初は，FM-8のユーザーとして，大変腹立たしかったFM-7も，FM-8の隣りに並んでいるいまでは，その性能のすごさに驚かされています．使い方次第で，FM-8との連携プレイもこなしてしまうのですね．

なお，本プログラムで作ったDOSモードDISK-BASICの入ったディスケットを使ってブートするときは，DIPスイッチの変更を忘れないでください(ROM/DISKモードのままでも，ブートしますが)．

◘参考文献

1) FM-7/8活用研究

2) I/O誌 FM-7/8に関する号

図2　実行時のハードコピー

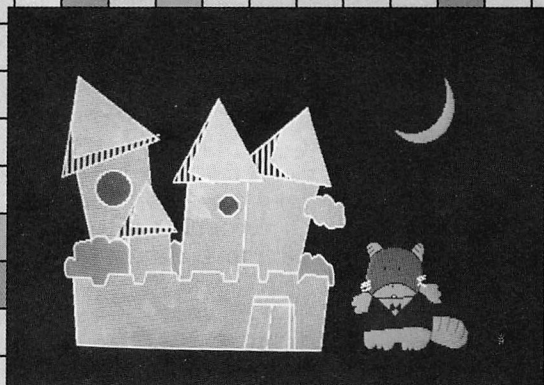
```
LO.M"DOS",,R

Set the All RAM disket on Drive 0

If Ready then Hit RETURN key !!

Ready
FILES

DOS-8      0 B S 1

134 Clusters Free

Ready
HARDC
```

All RAM DISK BASIC作成プログラム

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6000 20 28 0A 00 61 BC 00 00 00 00 09 00 00 00 00 00 :78
6010 00 00 8E 60 02 20 03 8E 60 0A E7 04 5F 81 10 2F :15
6020 03 80 10 5C ED 05 6E 9F FB FA BD 61 19 8E 61 BC :C5
6030 BF 60 0C CC 0F 00 FD 61 62 CC 01 02 FD 61 64 CC :23
6040 6E 00 FD 62 C4 CC 02 01 FD 62 C6 CC 00 00 FD 62 :B0
6050 C8 8E 70 00 10 8E 6E 00 C6 38 B6 FB FA 81 F2 27 :15
6060 05 1F 21 5A 8B 04 BF 63 28 F7 61 6C 80 EF B7 61 :C3
6070 67 FC 61 62 8D 9C FC 61 64 8D 9C 7C 61 64 7C 61 :57
6080 62 B6 61 62 81 21 25 E9 8E 80 00 BF 60 0C FC 61 :21
6090 64 8D 84 7C 61 64 B6 61 64 81 21 25 08 86 01 B7 :3E
60A0 61 64 7C 61 65 CC 01 00 F3 60 0C FD 60 0C 10 83 :2F
60B0 FC 00 26 DA 8E 61 BC BF 60 0C 5F 86 FF A7 80 5A :37
60C0 26 FB CC 04 01 FD 61 62 FC 61 62 BD 60 17 7C 61 :82
60D0 62 B6 61 62 81 21 25 F0 8E 61 C1 C6 12 6A 80 5A :5E
60E0 26 FB 8E 61 BC 6F 84 CC 01 01 BD 60 17 8E 61 BC :6C
60F0 C6 20 6F 80 5A 26 FB 8E 61 66 10 8E 61 BC C6 08 :2E
---------------------------------------------------------
Sum: 1B 24 54 06 B8 40 36 08 3D 84 A3 EE 01 54 A7 76 :93

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6100 A6 80 A7 A0 5A 26 F9 CC 04 01 BD 60 17 8E 62 C0 :9B
6110 BF 60 0C CC 01 00 7E 60 17 8D 37 8E 61 6E 10 8E :AC
6120 FC 82 C6 4E A6 80 A7 A0 5A 26 F9 7F FD 05 8D 22 :A8
6130 CC 29 00 FD FC 82 7F FD 05 8D 17 86 80 B7 FC 80 :CE
6140 B6 FC 83 7F FD 05 7D 03 13 26 05 81 0D 26 DF 39 :40
6150 35 90 B6 FD 05 2B FB 86 80 B7 FD 05 B6 FD 05 2A :44
6160 FB 39 00 00 00 00 11 04 44 4F 53 2D 38 09 03 4C :EC
6170 0D 0A 0A 11 04 53 65 74 20 74 68 65 20 41 6C 6C :FC
6180 20 52 41 4D 20 64 69 73 6B 65 74 20 6F 6E 20 44 :05
6190 72 69 76 65 20 30 0D 0A 0A 11 07 49 66 20 52 65 :C5
61A0 61 64 79 20 74 68 65 6E 20 48 69 74 20 52 45 54 :5D
61B0 55 52 4E 20 6B 65 79 20 21 21 0A 0D 00 00 00 00 :D7
61C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
61D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
61E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
61F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: 68 CB 3A 36 22 0C DF D5 27 C0 AF F5 05 05 05 08 :27

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
62C0 20 18 0A 00 6E 00 02 01 00 00 0A 00 03 98 00 03 :5B
62D0 00 00 08 00 00 00 00 00 00 00 10 CE 03 00 86 FD :6C
62E0 1F 8B 30 8C E5 8D 48 AE 02 A6 84 81 53 26 61 C6 :1B
62F0 8E 30 8C CE 34 04 BD FE 08 4D 27 06 81 0A 27 50 :8F
---------------------------------------------------------
Sum: CD D3 CE 5A 87 91 07 AD 0A F3 C5 55 DA C8 0E 16 :71

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6300 20 2D CC 01 00 E3 02 ED 02 A6 05 4C 81 10 2F 0D :B2
6310 6D 06 27 05 6C 04 6F 06 8C 6C 06 86 01 A7 05 35 :EA
6320 04 5A 26 D0 4F 1F 8B 8E 6E 00 43 6E 9F FB FE C6 :58
6330 05 34 04 34 10 30 8C 9A BD FE 02 35 10 81 0A 27 :8B
6340 0F BD FE 08 35 04 4D 27 0E 81 0A 27 03 5A 26 E1 :A3
6350 86 81 B7 FD 03 20 FE 39 00 00 00 00 00 00 00 00 :15
---------------------------------------------------------
Sum: 2B FF D2 0F 03 5A D3 7B C7 91 5A 9C 34 8D 62 10 :37
```

注）$6200～$62BFまで00にしてください．

FM-11 CP/M-86用C言語を使ったお絵描きプログラム
グラフィックエディタ

■Yes

これはFM-11のCP/M-86上で走る，グラフィック・エディタであり，C言語で記述されています．FM-11は高度なグラフィック機能を持ったマシンですが，残念ながらそれを生かしたソフトはあまりでていないようです．

単なる事務用のビジネス・マシンとして終わらせるのは，非常にもったいないことです．他メーカーの機種と比べても，グラフィック機能は充実しており，またDISPLAY SUBSYSTEMが公開されているので，BASIC 以外のOSからでも簡単にグラフィック機能を使うことができます．このグラフィック・エディタの場合は，Cで書かれていますが，Cの言語仕様には入出力機能すらなく（ただし，標準ライブラリ関数に含まれる)，ましてグラフィック機能は備わっていないため，まず，グラフィック・ライブラリの作成を行なわなければなりません．

私の場合，既にBASICとほぼ同機能を有するグラフィック・ライブラリを作成済だったので，作業は楽に進みました．

グラフィック・ライブラリについて

F-BIOS直接コールを使って，ディスプレイ・サブシステムをコントロールします．RCBインタフェイスのSUBINないし，SUBOUTです。RCBインタフェイスは，ソフトウェア・インタラプトの221(F-BIOSのシステム・コールNo.)で呼出します．

私の使用しているCコンパイラには，sysintというライブラリ関数が含まれており，これによって，アセンブラを使わずに，グラフィック・ライブラリの作成が可能になりました．

もし，こういった関数がない場合には，この部分をアセンブラで記述すれば良いことになります．注意する点としては，80系では2バイトの整数が，上位バイトと下位バイトが逆さまになって入っているということです．座標パラメータなどをユーザーバッファに作ってサブシステムに渡す際には，上位バイトと下位バイトを入れ換えておかないとダメです．

グラフィック・エディタについて

プログラムについて

形式としては，中間コード・インタプリタ形式をとっており，それぞれのグラフィック・コマンドに中間コードがつけられています．出力ファイルは，この中間コードのファイルと，Cのソース・ファイルの2種類になっています。

そして，このグラフィック・エディタは中間ファイルの方を読み込んで，エディットを行なうことになります．Cのソース・ファイルの方は当然そのままでは実行できませんが，これをコンパイルしてやれば，より高速の実行が可能になります．したがって，すべてのエディットが終了した段階で，Cのソース・ファイルをコンパイルしてやれば良いわけです．また，ランタイム・ルーチンも用意してあるので，中間コードによる実行も可能です．

操作はグラフィック・カーソルを，カーソル・コントロール・キーによって移動させ，位置決めをしてから，グラフィック・コマンドを入力する方式です．グラフィック・コマンドは標準的なものですが，中間色ペイントは 259色のサンプルが表示され，その中から選べるようになっています．また，この際（サンプル表示のとき）は，一種のマルチ・ウインドウ方式になっています．

これはFM-11のマルチ・ページ機能を生かしたものでサンプルを表示するフィールドのVRAMエリアを，サブシステムのCOPY GRAPHIC VRAMコマンドを使って，使用していないページのVRAMエリアに転送，退避させています．これは，単なるメモリのブロック転送だけに，かなり高速です．ただし，1バイト単位での転送になりますが．

コマンド・ラインも同様に使用していないVRAMに退避させているため，画面は全画面フルに使えます．また，現在までに行なった任意のコマンドを削除したり，どこにでも挿入することが可能なので，試行錯誤的に色決めをしたりするのに都合が良くなっています．

入力可能なコマンド数は，グラフィック・コマンドの種類によってパラメータの数が異なるため（たとえばPAINTは最も多く，PSETなどは少ない)，一概には言えませんが，通常4,000コマンドぐらいは入ります．フリーエリアが不足してきて危くなったときには，強制終了するようになっています．

本グラフィック・エディタの開発期間は，構想からデバックまで入れて，2週間ほどと，Cの開発効率の良さを示してくれました．ただ，前述したように，グラフィック・ライブラリが作成済だったので，その分労力が少なくて済んだようです．

操作法について

先に述べたように，すべてC言語で記述してあります．このうちのGEDITがグラフィック･エディタのメインです．これはCの標準ライブラリ，グラフィック・ライブラリとリンクさせて“CMD” ファイルを作ります．

TCOLORは中間色ペイントのためのタイルストリング・パターンが含まれるファイルで，GEDITを走らせると，まずこれを読み込みます．これはDDT86などで入れてください．このダンプにはヘッダが付いています．

RUNはランタイム・ルーチンで，中間ファイルを読み込んで実行します．これも，“CMD”のコマンド･ファイルにしておきます．

なお，グラフィック・ライブラリはFM-11用なので，その他のマシンで使うときは，そのマシンに合わせて書いてください．

さて具体的な操作ですが，仮にGEDITのディスクがＡに入っているとします．

```
A>GEDIT␣ファイル名(8文字以下，拡張子はつけない)
```

とすると，GEDITが起動します．画面がクリアされ，既存のファイルならば中間ファイルを読み込んで，前回描いたところまでトレースして入力待ちになります．新しいファイルならば，そのまま入力待ちになります．コマンド・ラインは水色で表示されます．このモードで使えるコマンドは，以下の通りです．入力は大文字でも小文字でも構いません(*n*は数字を表わします)．

I:インサート・モードに入る（つまり画像の入力モード)．

Tn:nステップ・トレースし，CPを移動する．*n*が負なら*n*ステップ戻る．

Dn:現在のステップから*n*ステップ前までのコマンドを削除する．

TB:最初に戻る．

TE:最後までトレースする．

E:エディットを終了し，ディスクに結果をセーブする．

QUIT:エディットの中止，結果はセーブしない．

T(トレース）コマンドを使って，任意の点にCPを移動し，**I**(インサート）コマンドによってグラフィック・コマンドを挿入する形になります（感覚的には，CP/Mのテキスト･エディタと同じです)．したがって，実際のグラフィック･コマンドの入力は，インサート･モード下で行なわれます．

インサート・モードに入ると，コマンド・ラインは緑色になり，画面上に＋印のグラフィック・カーソルと起点座標を示す□印が現われます．このグラフィック・カーソルをカーソル・コントロール・キーによって動かし，位置を決めてキーを押すと座標データが入力されます．

通常グラフィック・カーソルは１ドットずつ動きますが，速く動かしたいときには数字キーを押してから，カーソル・キーを使えば，そのドット数ずつ動きます．また，シフト・キーとカーソル・キーを併用すると，20ドットずつ動きます．

こうして起点座標（□印）と終点座標が入力され，コマンド・ライン上に*X*1, *Y*1, *X*2, *Y*2として表示されます．次にグラフィック・コマンドの入力です．なお，色コードは，BASICとコンパチになっています．

Ln(ライン）…起点～終点を結ぶ直線．*n*は表示色を示す．省略すると直前の表示色になる．直前の表示色はコマンド・ラインに表示されている．

Bn(ボックス）…起点と終点を対角線とする箱を描く，*n*については上に同じ．

BF*n*(ボックス・フィル）…起点と終点を対角線とする箱を描き，内部を塗りつぶす．*n*は同上．

PO*n*(ポイント）…終点に点を打つ．*n*は同上．

CR*n*1, *n*2(サークル）…*n*1を半径として，終点を中心とする円を描く．*n*2は色．

P*n*0, *n*1, *n*2, *n*3, *n*4, *n*5, *n*6, *n*7（ペイント)…終点位置からペイント．*n*0は塗色．*n*1～*n*7は境界色．境界色を省略すると，背景色の部分のみ塗りつぶす．

TP*n*1, *n*2, *n*3, *n*4, *n*5, *n*6, *n*7(タイル・ペイント）…終点位置から中間色塗り．*n*1～*n*7は境界色（ペイントと同じ)．キャリッジ・リターンすると，右端に色見本が表示される．色見本の先を見たいときは，＋を押してから⏎，戻りたいときは⊟，⏎です．259色ある．欲しい色がきまったら，そのNo.を入力する．

M(ムーブ）…単に起点を終点の位置まで移動．

GC*n*…グラフィック・カーソルの色を変える．*n*は色コード．

CAN(キャンセル）…何もせずに，もう一度グラフィックカーソル・モードになる．

CLS…画面クリア．

インサート・モードから脱けるには ESC ＋⏎を入力します．なお，コマンド・ライン上のCP, EPというのはEPが既に入力された最後のコマンド・ナンバーで，CPは現在エディットを行なっているコマンド・ナンバーです．これが一致しているときは，最後につけ加えている場合です．(つまり，テキスト・エディタの行番号のようなものです．CPを移動させるには，前述の通り**T**(トレース)コマンドを使います．

エンド・コマンドによりエディットを終えると，前述した通り，２つのファイルが出力されます．Cのソース・ファイルは，コンパイルしないと実行できませんが，中間ファイルを実行するには，**run.CMD**という，ランタイム・ルーチンを使えば実行できます．

```
A>run␣ファイル名（拡張子なし）
```

で実行します．

Cのソース・ファイルをコンパイルするには，Ｃコンパイラ，およびグラフィック・ライブラリが必要になります．

最後に

ここで使ったのは，CI社のＣコンパイラですが，現在CP/M-86ないし，MS-DOSあるいは，CP/M上で走るＣコンパイラは結構出回っているようです．価格が高いのが難点ですが，Ｃは単なるシステム記述言語ではなく，非常に汎用性のある言語であり，それなりの利用価値はあるでしょう．グラフィックなど行なうことを考えると，ハードに密着した関数，OSとのインターフェイスを提供する関数を含んでいるＣコンパイラの方が，いろいろと便利だと言えます．また，アセンブラとのリンクが可能かどうかも重要です．今後も，FM-11 CP/M-86上で，ゲームなどを作成しようと思っています．また，このグラフィック・インタプリタを使った，グラフィックが可能なPROLOGなどの作成も考えています．

width(w,line)
unsigned char w,line;
w ：画面表示文字幅．40ないし80．
line：表示行数．20ないし25．
どちらもパラメータが範囲外のときは現状維持．

console(sl,ns,pf)
unsigned char sl,ns,pf
sl：スクロール開始行(0～表示行数)．
ns：スクロール行数．
pf：ファンクション・キー表示スイッチ（1で表示，0で表示せず）．

cls(n)
unsigned char n;
n：消去モード
0…グラフィック画面とテキスト画面全部消去．
1…スクロール・モードのテキスト画面を消去．
2…ページ・モード1のテキスト画面を消去．
3…ページ・モード2のテキスト画面を消去．
4…テキスト画面のみ全画面消去．
5…グラフィック画面のみ全画面消去

locate(x1,y1)
int x1,y1
x1→カーソル座標(0～79)
y1→カーソル座標(0～24)

beep( )
ビープ音の発生．

color(fgcolor,bcolor)
int tgcolor,bcolor;
文字表示色および背景色の設定．
fgcolor：表示色．
bcolor：背景色はグラフィック画面の消去を行なわないと変わらない)．

atribute(n)
unsigned char n;
表示文字のアトリビュートを設定する．
n：
0…ノーマル
1…リバース
2…ブリンク
3…リバース・ブリンク
4…インテンシティ
5…インテンシティ・リバース
6…インテンシティ・ブリンク
7：インテンシティ・リバース・ブリンク

linec(X1,Y1,X2,Y2,c,color,fig)
int X1,Y1,X2,Y2,color,fig;
unsigned char c;
キャラクタで線や四角形を描く．
X1,Y1,：始点のX座標，Y座標(キャラクタ座標)．
X2,Y2：終点のX座標，Y座標(キャラクタ座標)．
C：表示に使う文字の値．
color：表示色
fig：L(0)…線
B(1)…四角形
BF(2)…四角形塗りつぶし．

line(X1,Y1,X2,Y2,func,color,fig)
int X1,Y1,X2,Y2,func,color,fig;
X1,Y1：始点のX座標，Y座標(グラフィック座標)．
X2,Y2：終点のX座標，Y座標(グラフィック座標)．
func：機能：
PSET(0)
PRESET(1)
OR (2)
AND (3)
XOR (4)
fig：L (0)
B (1)
BF(2)

sline(X1,Y1,X2,Y2,func,color,fig,style)
int X1,Y1,X2,Y2,func,color,fig,style;
スタイル付直線
style：16ビットのドット・パターン．この繰り返しで線を引く．
他のパラメータはlineと同じ．

pset(X1,Y1,color,func)
int X1,Y1,color,func;
X1,Y1→表示座標．
color→表示色．
func→機能．(PSET(0)～XOR(4))．

paint(X1,Y1,color,bnom,b1,b1,b2,b3,b4,b5,b6,b7)
int X1,Y1,color,bnom,b0,b1,b2,b3,b4,b5,b6,b7;
X1,Y1：塗り始める位置．
color：表示色．
bnom：境界色の数．
b0～b7：境界色(bnomで指定した数だけ有効)．

screen(mode,actpage,dispage,actbank dispbank)
int mode,actpage,dispage,actbank,dispbank;
画面モードおよびアクティブ・ページ，ディスプレイ・ページ，アクティブ・バンク，ディスプレイ・バンクの設定．
mode：画面モード．
0…カラーモード(640×200)
1…カラーモード(640×400)
2…単色モード(640×200)
3…単色モード(640×400)
4…グラフィックスクロール・モード(640×200)
5…グラフィックスクロール・モード(640×400)
＊カラーモードではアクティブ・バンク，ディスプレイ・バンクの指定は無視される．

tpaint(X1,Y1,tn,t1,t2,t3,t4,bnom,b0,b1,b2,b3,b4,b5,b6,b7)
int X1,Y1;
unsigned char *t1,*t2,*t3,*t4;
unsigned char tn,bnom,b0,b1,b2,b3,b4,b5,b6,b7;
タイルストリング・パターンによるペイント．
tn→タイルストリングの数(1～4)．
t1～t4 3バイトのタイルストリングへのポインタ．
後の引数はペイントと同じ．

unsigned char *get_ga(X1,Y1,X2,Y2)
int X1,Y1,X2,Y2;
X1,Y1,X2,Y2を対角線とする枠内のBANK0～2の画面データを読みとり，メモリ上に拡納する．
return値は取り込んだドット情報の先頭のポインタ．
ドット情報のうち最初の2バイトにバイト数．

put_ga(X1,Y1,X2,Y2,p,func)
int X1,Y1,X2,Y2,func;
unsigned chan *p;
get_gaで取り込んだドット情報に従って，表示する．
P：ドット情報の拡納されているメモリの先頭のポインタ．
func：機能

unsigned char *get_g(X1,Y1,X2,c0,c1,c2,c3,c4,c5,c6,c7)
int X1,Y1,X2,Y2;
unsigned char c0,c1,c2,c3,c4,c5,c6,c7;
X1,Y1,X2,Y2の枠内の指定した

グラフィック・ライブラリ

色のドット情報を取り込む.
return値は取り込んだドット情報の先頭のポインタ.
c0〜c7：指定する色で"#"で終了になる.

put_g(X1,Y1,X2,Y2,P,func,color)

```
int X1,Y1,X2,Y2;
unsigned chan *p
unsigned chan func,color;
```

get_g( )で取り込んだドット情報に従って，指定色，機能で描画する.
P：ドット情報の先頭ポインタ.
func：機能.
color：表示色.

symbol(X1,Y1,s,yoko,tate,color,angle,func)

```
int X1,Y1,yoko,tate,color,func;
unsighed char angle;
char *s;
```

文字列を指定した倍率，角度で表示.
X1,Y1：左上の表示座標（グラフィック座標）.
S：文字配列へのポインタ.
yoko：横倍率
tate：縦倍率.
color：表示色.
angle：角度コード.
- 0…ノーマル(8)
- 1…90°左回転(9)
- 2…180°左回転(10)
- 3…270°左回転(11)

→8×16の文字パターンの拡大.
func：機能.

roll(tate,yoko)

```
int tate,yoko;
```

グラフィック画面のスクロール.
tate…上下方向ドット数
正→上に
負→下に
(×200モード)(×400
200
〜
−200
(×400モード)
400
〜
−400
yoko…横のドット数
正→左
負→右
640
〜
−640
400
〜
−640 →16の倍数.

タイトル・ストリング・パターン　ダンプ・リスト(tcolor. dat)

```
0AD0:0000 03 01 00 00 00 00 00 00 55 00 00 AA 00 00 00 55
0AD0:0010 00 00 AA 00 55 55 00 AA AA 00 00 00 55 00 00 AA
0AD0:0020 55 00 55 AA 00 AA 00 55 55 00 AA AA 55 55 55 AA
0AD0:0030 AA AA FF 00 00 FF 00 00 AA 55 00 55 AA 00 FF 55
0AD0:0040 00 FF AA 00 AA 00 55 55 00 AA FF 00 55 FF 00 AA
0AD0:0050 AA 55 55 55 AA AA FF 55 55 FF AA AA 00 FF 00 00
0AD0:0060 FF 00 55 FF 00 AA FF 00 00 AA 55 00 55 AA 55 AA
0AD0:0070 55 AA 55 AA 00 FF 55 00 FF AA 55 FF 55 AA FF AA
0AD0:0080 FF FF 00 FF FF 00 AA AA 55 55 55 AA FF AA 55 FF
0AD0:0090 55 AA AA FF 55 55 FF AA FF FF 55 FF FF AA 00 00
0AD0:00A0 FF 00 00 FF 55 00 FF AA 00 FF 00 55 FF 00 AA FF
0AD0:00B0 55 55 FF AA AA FF FF 00 FF FF 00 FF AA 55 FF 55
0AD0:00C0 AA FF FF 55 FF FF AA FF 00 FF FF 00 FF FF 55 FF
0AD0:00D0 FF AA FF FF 11 00 00 44 00 00 00 11 00 00 44 00
0AD0:00E0 11 11 00 44 44 00 00 00 11 00 00 44 11 00 11 44
0AD0:00F0 00 44 00 11 11 00 44 44 11 11 11 44 44 44 44 11
0AD0:0100 00 11 44 00 55 11 00 55 44 00 44 00 11 11 00 44
0AD0:0110 55 00 11 55 00 44 44 11 11 11 44 44 55 11 11 55
0AD0:0120 44 44 11 55 00 44 55 00 00 44 11 00 11 44 11 44
0AD0:0130 11 44 11 44 00 55 11 00 55 44 11 55 11 44 55 44
0AD0:0140 44 44 11 11 11 44 55 44 11 55 11 44 44 55 11 11
0AD0:0150 55 44 55 55 11 55 55 44 11 00 55 44 00 55 00 11
0AD0:0160 55 00 44 55 11 11 55 44 44 55 44 11 55 11 44 55
0AD0:0170 55 11 55 55 44 55 11 55 55 44 55 55 BB 00 00 EE
0AD0:0180 00 00 AA 11 00 AA 44 00 BB 11 00 EE 44 00 AA 00
0AD0:0190 11 AA 00 44 BB 00 11 EE 00 44 AA 11 11 AA 44 44
0AD0:01A0 BB 11 11 EE 44 44 EE 11 00 BB 44 00 FF 11 00 FF
0AD0:01B0 44 00 EE 00 11 BB 00 44 FF 00 11 FF 00 44 EE 11
0AD0:01C0 11 BB 44 44 FF 11 11 FF 44 44 BB 55 00 EE 55 00
0AD0:01D0 AA 44 11 AA 11 44 BB 44 11 EE 11 44 AA 55 11 AA
0AD0:01E0 55 44 BB 55 11 EE 55 44 EE 44 11 BB 11 44 FF 44
0AD0:01F0 11 FF 11 44 EE 55 11 BB 55 44 FF 55 11 FF 55 44
0AD0:0200 BB 00 55 EE 00 55 AA 11 55 AA 44 55 BB 11 55 EE
0AD0:0210 44 55 EE 11 55 BB 44 55 FF 11 55 FF 44 55 BB 55
0AD0:0220 55 EE 55 55 11 AA 00 44 AA 00 00 BB 00 00 EE 00
0AD0:0230 11 BB 00 44 EE 00 00 AA 11 00 AA 44 11 AA 11 44
0AD0:0240 AA 44 00 BB 11 00 EE 44 11 BB 11 44 EE 44 44 BB
0AD0:0250 00 11 EE 00 55 BB 00 55 EE 00 44 AA 11 11 AA 44
0AD0:0260 55 AA 11 55 AA 44 44 BB 11 11 EE 44 55 BB 11 55
0AD0:0270 EE 44 11 FF 00 44 FF 00 00 EE 11 00 BB 44 11 EE
0AD0:0280 11 44 BB 44 00 FF 11 00 FF 44 11 FF 11 44 FF 44
0AD0:0290 44 EE 11 11 BB 44 55 EE 11 55 BB 44 44 FF 11 11
0AD0:02A0 FF 44 55 FF 11 55 FF 44 11 AA 55 44 AA 55 00 BB
0AD0:02B0 55 00 EE 55 11 BB 55 44 EE 55 44 BB 55 11 EE 55
0AD0:02C0 55 BB 55 55 EE 55 11 FF 55 44 FF 55 BB AA 00 EE
0AD0:02D0 AA 00 AA BB 00 AA EE 00 BB BB 00 EE EE 00 AA AA
0AD0:02E0 11 AA AA 44 BB AA 11 EE AA 44 AA BB 11 AA EE 44
0AD0:02F0 BB BB 11 EE EE 44 EE BB 00 BB EE 00 FF BB 00 FF
0AD0:0300 EE 00 EE AA 11 BB AA 44 FF AA 11 FF AA 44 EE BB
0AD0:0310 11 BB EE 44 FF BB 11 FF EE 44 BB FF 00 EE FF 00
0AD0:0320 AA EE 11 AA BB 44 BB EE 11 EE BB 44 AA FF 11 AA
0AD0:0330 FF 44 BB FF 11 EE FF 44 EE EE 11 BB BB 44 FF EE
0AD0:0340 11 FF BB 44 EE FF 11 BB FF 44 FF FF 11 FF FF 44
0AD0:0350 BB AA 55 EE AA 55 AA BB 55 AA EE 55 BB BB 55 EE
0AD0:0360 EE 55 EE BB 55 BB EE 55 FF BB 55 FF EE 55 BB FF
0AD0:0370 55 EE FF 55 11 00 AA 44 00 AA 00 11 AA 00 44 AA
0AD0:0380 11 11 AA 44 44 AA 00 00 BB 00 00 EE 11 00 BB 44
0AD0:0390 00 EE 00 11 BB 00 44 EE 11 11 BB 44 44 EE 44 11
0AD0:03A0 AA 11 44 AA 55 11 AA 55 44 AA 44 00 BB 11 00 EE
0AD0:03B0 55 00 BB 55 00 EE 44 11 BB 11 44 EE 55 11 BB 55
0AD0:03C0 44 EE 11 55 AA 44 55 AA 00 44 BB 00 11 EE 11 44
0AD0:03D0 BB 44 11 EE 00 55 BB 00 55 EE 11 55 BB 44 55 EE
0AD0:03E0 44 44 BB 11 11 EE 55 44 BB 55 11 EE 44 55 BB 11
0AD0:03F0 55 EE 55 55 BB 55 55 EE 11 00 FF 44 00 FF 00 11
0AD0:0400 FF 00 44 FF 11 11 FF 44 44 FF 44 11 FF 11 44 FF
0AD0:0410 55 11 FF 55 44 FF 11 55 FF 44 55 FF BB 00 AA EE
0AD0:0420 00 AA AA 11 AA AA 44 AA BB 11 AA EE 44 AA AA 00
0AD0:0430 BB AA 00 EE BB 00 BB EE 00 EE AA 11 BB AA 44 EE
0AD0:0440 BB 11 BB EE 44 EE EE 11 AA BB 44 AA FF 11 AA FF
0AD0:0450 44 AA EE 00 BB BB 00 EE FF 00 BB FF 00 EE EE 11
0AD0:0460 BB BB 44 EE FF 11 BB FF 44 EE BB 55 AA EE 55 AA
0AD0:0470 AA 44 BB AA 11 EE BB 44 BB EE 11 EE AA 55 BB AA
0AD0:0480 55 EE BB 55 BB EE 55 EE EE 44 BB BB 11 EE FF 44
0AD0:0490 BB FF 11 EE EE 55 BB BB 55 EE FF 55 BB FF 55 EE
0AD0:04A0 BB 00 FF EE 00 FF AA 11 FF AA 44 FF BB 11 FF EE
0AD0:04B0 44 FF EE 11 FF BB 44 FF FF 11 FF FF 44 FF BB 55
0AD0:04C0 FF EE 55 FF 11 AA AA 44 AA AA 00 BB AA 00 EE AA
0AD0:04D0 11 BB AA 44 EE AA 00 AA BB 00 AA EE 11 AA BB 44
0AD0:04E0 AA EE 00 BB BB 00 EE EE 11 BB BB 44 EE EE 44 BB
0AD0:04F0 AA 11 EE AA 55 BB AA 55 EE AA 44 AA BB 11 AA EE
0AD0:0500 55 AA BB 55 AA EE 44 BB BB 11 EE EE 55 BB BB 55
0AD0:0510 EE EE 11 FF AA 44 FF AA 00 EE BB 00 BB EE 11 EE
0AD0:0520 BB 44 BB EE 00 FF BB 00 FF EE 11 FF BB 44 FF EE
0AD0:0530 44 EE BB 11 BB EE 55 EE BB 55 BB EE 44 FF BB 11
0AD0:0540 FF EE 55 FF BB 55 FF EE 11 AA FF 44 AA FF 00 BB
0AD0:0550 FF 00 EE FF 11 BB FF 44 EE FF 44 BB FF 11 EE FF
0AD0:0560 55 BB FF 55 EE FF 11 FF FF 44 FF FF BB AA AA EE
0AD0:0570 AA AA AA BB AA AA EE AA BB BB AA EE EE AA AA AA
0AD0:0580 BB AA AA EE BB AA BB EE AA EE AA BB BB AA EE EE
0AD0:0590 BB BB BB EE EE EE EE BB AA BB EE AA FF BB AA FF
0AD0:05A0 EE AA EE AA BB BB AA EE FF AA BB FF AA EE EE BB
0AD0:05B0 BB BB EE EE FF BB BB FF EE EE BB FF AA EE FF AA
0AD0:05C0 AA EE BB AA BB EE BB EE BB EE BB EE AA FF BB AA
0AD0:05D0 FF EE BB FF BB EE FF EE EE EE BB BB BB EE FF EE
0AD0:05E0 BB FF BB EE EE FF BB BB FF EE FF FF BB FF FF EE
0AD0:05F0 BB AA FF EE AA FF AA BB FF AA EE FF BB BB FF EE
0AD0:0600 EE FF EE BB FF BB EE FF FF BB FF FF EE FF BB FF
0AD0:0610 FF EE FF FF 1A BB 44 55 FF 11 55 FF 44 55 BB 55
0AD0:0620 55 EE 55 55 11 AA 00 44 AA 00 00 BB 00 00 EE 00
0AD0:0630 11 BB 00 44 EE 00 00 AA 11 00 AA 44 11 AA 11 44
0AD0:0640 AA 44 00 BB 11 00 EE 44 11 BB 11 44 EE 44 44 BB
0AD0:0650 00 11 EE 00 55 BB 00 55 EE 00 44 AA 11 11 AA 44
0AD0:0660 55 AA 11 55 AA 44 44 BB 11 11 EE 44 55 BB 11 55
0AD0:0670 EE 44 11 FF 00 44 FF 00 00 EE 11 00 BB 44 11 EE
```

スタンダードグラフィック・ライブラリ(stdgra. h)

```
/*  standard graphic libraray heading
*/

#define  PSET   0
#define  PRESET 1
#define  OR     2
#define  AND    3
#define  XOR    4

#define  L      0
#define  B      1
#define  BF     2
```

```
/*    graphic library   */

#include "stdgra.h"
#include "stdio.h"

unsigned char  usrbuf[128];

struct rcb
     {
          unsigned char  reqno, status;
          unsigned  usroff, usrseg;
          unsigned char  nofbyte, null, reserve, r_byte;
     } g_rcb = { 0,0,0,0,0,0,0,0 };

struct   /*  current console parameter aria  */
     {
          unsigned char cur_x, cur_y, atri, nc, nl, sl, ns, pf;
          unsigned  cf;
     } cons;

struct         /*  current vram status parameter   */
     {
           unsigned char  hard, act, disp;
           unsigned  page0, page1, page2, page3;
     } vram;

static struct
     {
           unsigned  gx, gy, bcolor;
     } current = { 0, 0, 0 };

struct regval
     {
          int  ax, bx, cx, dx, si, di, ds, es;
     } sr, rr;

width(w, line)
unsigned char w, line;
{
       read_cons();
       usrbuf[2] = 0x01;
       usrbuf[3] = cons.atri;
       usrbuf[4] = (w == 40 || w == 80)?  w : cons.nc;
       usrbuf[5] = (line == 20 || line == 25)?  line : cons.nl;
       usrbuf[6] = 0;
       usrbuf[7] = usrbuf[5];
       usrbuf[8] = cons.pf;
       usrbuf[9] = 1;

       g_rcb.nofbyte = 10;
       subout();
}

console(sl, ns, pf)
unsigned char  sl, ns, pf;
{
      read_cons();
      usrbuf[2] = 0x01;
      usrbuf[3] = cons.atri;
      usrbuf[4] = cons.nc;
      usrbuf[5] = cons.nl;
      usrbuf[6] = (sl>=0 && sl<=cons.nl)?  sl : 0;
      usrbuf[7] = (ns>=0 && ns<=cons.nl-usrbuf[6])? ns:cons.nl-usrbuf[6];
      usrbuf[8] = (pf != 1)?  0 : 1;
      usrbuf[9] = 0;

      g_rcb.nofbyte = 10;
      subout();
}

cls(n)
unsigned char n;
{
      n = (n>=0 && n<=5)?  n : 0;
      if (n==0 || n==5)
      {
          usrbuf[2] = 0x25;
          usrbuf[3] = current.bcolor;
          g_rcb.nofbyte = 4;
          subout();
      }
      if (n != 5)
      {
          read_cons();
          usrbuf[2] = 0x02;    /* console erase command code set */
          usrbuf[4] = cons.atri;  /* console atribute set */
          switch (n)
          {
```

```
		case 0:
			usrbuf[3] = 0;
			break;
		case 1:
			usrbuf[3] = 1;
			break;
		case 2:
			usrbuf[3] = 2;
			break;
		case 3:
			usrbuf[3] = 3;
			break;
		case 4:
			usrbuf[3] = 0;
			break;
		default:
			break;
		}
		g_rcb.nofbyte = 5;
		subout();
	}
}

locate(x1,y1)
int x1, y1;
{
	usrbuf[2] = 0x03;		/*  put string command code  */
	usrbuf[3] = 4;			/*  length of string */
	usrbuf[4] = 0x1b;
	usrbuf[5] = '=';
	usrbuf[6] = y1+0x20;		/*  offset 0x20   */
	usrbuf[7] = x1+0x20;

	g_rcb.nofbyte = 8;
	subout();
}

/*   BEEP   FUNCTION   */
beep()
{
	usrbuf[2] = 0x12;		/*  character out command code */
	usrbuf[3] = 0x07;		/*  beep code '07'  */
	g_rcb.nofbyte = 4;
	subout();
}

color(fgcolor, bcolor)
int  fgcolor, bcolor;
{
	read_cons();
	fgcolor |= (cons.atri & 0xf8); /* mask and set color bit */
	put_atri(fgcolor);

	current.bcolor = bcolor;		/* background color set */
}

atribute(n)
unsigned char n;
{
	read_cons();			/*  read current atribute  */
	n <<= 3;			/*  convert parameter      */
	n |= (cons.atri & 0x07);	/* set atribute bit */
	put_atri(n);
}

put_atri(atri)
unsigned char atri;
{
	usrbuf[2] = 0x03;		/* put string command code '0x03' */
	usrbuf[3] = 3;			/* string length  */
	usrbuf[4] = 0x1b;		/* escape code '0x1b' */
	usrbuf[5] = 'G';
	usrbuf[6] = atri;
	g_rcb.nofbyte = 7;
	subout();
}

linec(x1,y1,x2,y2,c,color,func)
int  x1, y1, x2, y2, color, func;
unsigned char c;
{
	read_cons();
	usrbuf[2] = 0x20;		/* charcter line command code */
	usrbuf[3] = (cons.atri & 0xf8) | color;
	usrbuf[4] =  c;
	usrbuf[6] = x1;
	usrbuf[8] = y1;
	usrbuf[10] = x2;
```

```
	usrbuf[12] = y2;
	usrbuf[13] = func;
	g_rcb.nofbyte = 14;
	subout();
}

line(x1, y1, x2, y2, func, color, fig)
int  x1, y1, x2, y2, func, color, fig;
{
	usrbuf[2] = 0x15;
	usrbuf[3] = color;
	usrbuf[4] = func;
	usrbuf[5] = gethigh(x1);
	usrbuf[6] = x1;
	usrbuf[7] = gethigh(y1);
	usrbuf[8] = y1;
	usrbuf[9] = gethigh(x2);
	usrbuf[10] = current.gx = x2;
	usrbuf[11] = gethigh(y2);
	usrbuf[12] = current.gy = y2;
	usrbuf[13] = fig;

	g_rcb.nofbyte = 16;
	subout();
}

sline(x1,y1,x2,y2,func,color,fig,style)
int x1,y1,x2,y2,func,color,fig,style;
{
	 usrbuf[2] = 0x15;
	 usrbuf[3] = color;
	 usrbuf[4] = func;
	 usrbuf[5] = gethigh(x1);
	 usrbuf[6] = x1;
	 usrbuf[7] = gethigh(y1);
	 usrbuf[8] = y1;
	 usrbuf[9] = gethigh(x2);
	 usrbuf[10] = current.gx = x2;
	 usrbuf[11] = gethigh(y2);
	 usrbuf[12] = current.gy = y2;
	 usrbuf[13] = (fig == B)? 4 : 3;
	 usrbuf[14] = gethigh(style);
	 usrbuf[16] = style;

	 g_rcb.nofbyte = 16;
	 subout();
}

pset(x1, y1, color, func)
int  x1, y1, color, func;
{
	 usrbuf[2] = 0x17;        /* point command_code 0x17  */
	 usrbuf[3] = 1;           /* number of points  */
	 usrbuf[4] = gethigh(x1);
	 usrbuf[5] = current.gx = x1;    /*  set current x   */
	 usrbuf[6] = gethigh(y1);
	 usrbuf[7] = current.gy = y1;    /*  set current y value */
	 usrbuf[8] = color;
	 usrbuf[9] = (func>=0 && func <=4)?  func : 0;

	 g_rcb.nofbyte = 10;
	 subout();
}

paint(x1,y1,color,bnom,b0,b1,b2,b3,b4,b5,b6,b7)
int x1,y1,color,bnom,b0,b1,b2,b3,b4,b5,b6,b7;
{
	 usrbuf[2] = 0x18;  /* paint command code set */
	 usrbuf[3] = gethigh(x1);
	 usrbuf[4] = x1;
	 usrbuf[5] = gethigh(y1);
	 usrbuf[6] = y1;
	 usrbuf[7] = color;
	 usrbuf[8] = bnom;  /* number of border color */
	 usrbuf[9] = b0;    usrbuf[10] = b1;
	 usrbuf[11] = b2;   usrbuf[12] = b3;
	 usrbuf[13] = b4;   usrbuf[14] = b5;
	 usrbuf[15] = b6;   usrbuf[16] = b7;  /* border color code */

	 g_rcb.nofbyte = 17;
	 subout();
}

tpaint(x1,y1,tn,t1,t2,t3,t4,bnom,b0,b1,b2,b3,b4,b5,b6,b7)
int  x1, y1;
unsigned char *t1, *t2, *t3, *t4;
unsigned char tn, bnom, b0, b1, b2, b3, b4, b5, b6, b7;
{
```

```
        usrbuf[2] = 0x18;
        usrbuf[3] = gethigh(x1);
        usrbuf[4] = x1;
        usrbuf[5] = gethigh(y1);
        usrbuf[6] = y1;
        usrbuf[7] = 0xff;                /*  tile string flag */
        usrbuf[8] = bnom;
        usrbuf[9] = b0;          usrbuf[10] = b1;
        usrbuf[11] = b2;         usrbuf[12] = b3;
        usrbuf[13] = b4;         usrbuf[14] = b5;
        usrbuf[15] = b6;         usrbuf[16] = b7;
        usrbuf[17] = tn;             /* number of tile-string */
        movmem(t1, &usrbuf[18], 3);
        movmem(t2, &usrbuf[21], 3);
        movmem(t3, &usrbuf[24], 3);
        movmem(t4, &usrbuf[27], 3);
        g_rcb.nofbyte = 30;
        subout();
}

screen(mode, actpage, dispage, actbank, dispbank)
int mode, actpage, dispage, actbank, dispbank;
{
        unsigned char  sf;

        sf = actpage;
        sf <<= 3;            /* 3-bit shift for page set  */
        sf += ( mode <= 1 )?   7 : actbank;
        usrbuf[2] = 0x21;     /*  select active vram command */
        usrbuf[3] = sf;       /*  select-flag  set */
        g_rcb.nofbyte = 4;
        subout();

        sf = dispage;
        sf <<= 3;
        sf += ( mode <= 1 )?  7 : dispbank;
        if (mode==1 || mode==3 || mode==5)
            sf |= 0x40;              /*  6th bit on for 400line display */
        else
            sf &= 0x1f;              /*  6th bit off 200 line display */
        usrbuf[2] = 0x22;     /*  select display vram command */
        usrbuf[3] = sf;
        g_rcb.nofbyte = 4;
        subout();

        if (mode==4 || mode==5)
        {
            usrbuf[2] = 0x0f;          /* set console flag command */
            usrbuf[3] = 14;            /* with graphic scroll bit */
        }
        else
        {
            usrbuf[2] = 0x0e;          /* clear console flag command */
            usrbuf[3] = 14;
        }
        g_rcb.nofbyte = 4;
        subout();
}


unsigned char *get_ga(x1,y1,x2,y2)
int  x1, y1, x2, y2;
{
       unsigned char *p;
       long  tate, yoko;
       unsigned  len, number, i;

       yoko = (x1 < x2)?  x2-x1+1 : x1-x2+1;
       tate = (y1 < y2)?  y2-y1+1 : y1-y2+1;
       p = alloc(len = (yoko*tate+7)/8*3+2);
       number = len-2;
       movmem(&number, p, sizeof(int));
       p += 2;
       usrbuf[2] = 0x1d;              /* put block 2 command code */
       usrbuf[3] = gethigh(x1);     usrbuf[4] = x1;
       usrbuf[5] = gethigh(y1);     usrbuf[6] = y1;
       usrbuf[7] = gethigh(x2);     usrbuf[8] = x2;
       usrbuf[9] = gethigh(y2);     usrbuf[10] = y2;
       g_rcb.nofbyte = 11;
       g_rcb.r_byte  = (number > 124)?  128 : number+4;
       subin();
       movmem(&usrbuf[4], p, usrbuf[3]);
       p += usrbuf[3];                 /*   count up 'p'  */

       g_rcb.nofbyte = 3;
       g_rcb.r_byte = 128;
       while ( usrbuf[1] >= 0x80 )      /*  continue flag is on */
       {
```

```
            usrbuf[2] = 0x64;               /* continue command set */
            sysint(221, &sr, &rr);
            movblock(&usrbuf[4],g_rcb.usrseg,p,g_rcb.usrseg,usrbuf[3]);
            p += usrbuf[3];             /*  count up 'p'  */
        }
        return(p -= len);
}

unsigned char *get_g(x1,y1,x2,y2,c0,c1,c2,c3,c4,c5,c6,c7)
int x1, y1, x2, y2;
unsigned char c0, c1, c2, c4, c5, c6, c7;
{
        unsigned char *p;
        long  tate, yoko;
        unsigned char cl[8];
        unsigned  i, len, number;

        cl[0] = c0;  cl[1] = c1;  cl[2] = c2;  cl[3] = c3;
        cl[4] = c4;  cl[5] = c5;  cl[6] = c6;  cl[7] = c7; /* color set */
        yoko = (x1 < x2)? x2-x1+1 : x1-x2+1;
        tate = (y1 < y2)? y2-y1+1 : y1-y2+1;
        p = alloc(len = (yoko*tate+7)/8+2);
        number = len - 2;
        movmem(&number, p, sizeof(int));
        p += 2;
        usrbuf[2] = 0x1b;           /* get block 1 command code */
        usrbuf[3] = gethigh(x1);        usrbuf[4] = x1;
        usrbuf[5] = gethigh(y1);        usrbuf[6] = y1;
        usrbuf[7] = gethigh(x2);        usrbuf[8] = x2;
        usrbuf[9] = gethigh(y2);        usrbuf[10] = y2;
        for (i = 0; i < 8 && cl[i] != '#'; ++i)
              usrbuf[12+i] = cl[i];
        usrbuf[11] = i ;          /* number of color code  */
        g_rcb.nofbyte = 12 + usrbuf[11];
        g_rcb.r_byte = (number > 124)?  128 : number+4;
        subin();
        movmem(&usrbuf[4], p, usrbuf[3]);
        p += usrbuf[3];

        g_rcb.nofbyte = 3;
        g_rcb.r_byte = 128;
        while(usrbuf[1] >= 0x80)            /*  continue flag is on */
        {
              usrbuf[2] = 0x64;             /*  continue command set */
              sysint(221, &sr, &rr);
              movmem(&usrbuf[4], p, usrbuf[3]);
              p += usrbuf[3];
        }
        return(p -= len);
}

put_ga(x1, y1, x2, y2, p, func)
int x1, y1, x2, y2, func;
unsigned char *p;
{
        unsigned  number, i, j;

        movmem(p, &number, sizeof(int));
        p += 2;

        usrbuf[1] = ( number > 114 )?  0x80 : 0;
        usrbuf[2] = 0x1e;
        usrbuf[3] = gethigh(x1);     usrbuf[4] = x1;
        usrbuf[5] = gethigh(y1);     usrbuf[6] = y1;
        usrbuf[7] = gethigh(x2);     usrbuf[8] = x2;
        usrbuf[9] = gethigh(y2);     usrbuf[10] = y2;
        usrbuf[12] = func;
        usrbuf[13] = (number > 114)?   114 : number;
        movmem(p, &usrbuf[14], usrbuf[13]);
        p += usrbuf[13];
        number -= usrbuf[13];
        g_rcb.nofbyte = usrbuf[13]+14;
        subout();

        j = number/124;               /*  repetition of continue  */
        usrbuf[2] = 0x64;            /*  continue command code 0x64 */
        if ( j )
        {
              usrbuf[1] = 0x80;            /*  still continue flag  */
              usrbuf[3] = 124;
              g_rcb.nofbyte = 128;
              while ( j-- )
              {
                    movmem(p,&usrbuf[4],124);
                    p += 124;
                    sysint(221, &sr, &rr);       /* direct call sysint */
              }
        }
```

```
    usrbuf[1] = 0x00;              /*  last data (not continue) */
    usrbuf[3] = number % 124;       /*  remain byte  */
    g_rcb.nofbyte = 4+usrbuf[3];
    movmem(p, &usrbuf[4], usrbuf[3]);
    sysint(221, &sr, &rr);
}

put_g(x1, y1, x2, y2, p, func, color)
int x1, y1, x2, y2;
unsigned char *p, func, color;
{
      unsigned  number, i, j;

      movmem(p, &number, sizeof(int));
      p += 2;

      usrbuf[1] = ( number > 114 )?  0x80 : 0; /* continue? */
      usrbuf[2] = 0x1c;               /* put block 1 command */
      usrbuf[3] = gethigh(x1);  usrbuf[4] = x1;
      usrbuf[5] = gethigh(y1);  usrbuf[6] = y1;
      usrbuf[7] = gethigh(x2);  usrbuf[8] = x2;
      usrbuf[9] = gethigh(y2);  usrbuf[10] = y2;
      usrbuf[11] = color;
      usrbuf[12] = func;
      usrbuf[13] = ( number > 114 )?  114 : number;

      movmem(p, &usrbuf[14], usrbuf[13]);
      p += usrbuf[13];
      number -= usrbuf[13];
      g_rcb.nofbyte = usrbuf[13] + 14;
      subout();

      j = number / 124;
      usrbuf[2] = 0x64;
      if ( j )
      {
           usrbuf[1] = 0x80;
           usrbuf[3] = 124;
           g_rcb.nofbyte = 128;
           while( j-- )
           {
                 movmem(p, &usrbuf[4], 124);
                 p += 124;
                 sysint(221, &sr, &rr);
           }
      }
      usrbuf[1] = 0x00;
      usrbuf[3] = number % 124;
      g_rcb.nofbyte = 4 + usrbuf[3];
      movmem(p, &usrbuf[4], usrbuf[3]);
      sysint(221, &sr, &rr);
}

symbol(x1, y1, s, yoko, tate, color, angle, func)
int x1, y1, yoko, tate, color, func;
unsigned char  angle;
char *s;
{
      int  i = 0;

      if (angle >= 8)
          angle |= 0x80;          /*  8*16 dot font  */
      else
          angle &= 0x03;          /*  8*8 dot font   */
      usrbuf[2] = 0x19;           /*  symbol command code '0x19' */
      usrbuf[3] = color;
      usrbuf[4] = func;
      usrbuf[5] = angle;
      usrbuf[6] = yoko;
      usrbuf[7] = tate;
      usrbuf[8] = gethigh(x1);    usrbuf[9] = x1;
      usrbuf[10] = gethigh(y1);   usrbuf[11] = y1;
      usrbuf[12] = strlen(s);     /*  string length set */
      while ( *s )
         usrbuf[13+i++] = *s++;   /*  set char to usrbuf */

      g_rcb.nofbyte = ( i < 80 )?  (i+13) : 93;   /* 80 char limit */
      subout();
}

roll(tate, yoko)
int  tate, yoko;
{
     int  addr;              /* graphic start address data */

     addr = 0x50 * tate + 0x02 * (yoko/16);
     usrbuf[2] = 0x26;       /* set graphic start address command */
     usrbuf[3] = 1;
```

```
        usrbuf[4] = gethigh(addr);
        usrbuf[5] = addr;
        g_rcb.nofbyte = 6;
        subout();
}

#define ASP 45

circle(x, y, r, col)
int x, y, r, col;
{

    unsigned long x1, y1;
    int  ox, oy, xi, yi;
    unsigned long i;
    ox = 0;    oy = r * 450L / 1000;
    for (i=0;  i<=100;  i+=5)
    {
        x1 = 1000000*2*i/(10000+i*i)*r;
        xi = x1 / 10000;

        y1 = 100000*(10000-i*i)/(10000+i*i)*r * ASP;
        yi = y1 / 10000000;

        line(x+ox, y+oy, x+xi, y+yi, PSET, col, L);
        line(x+ox, y-oy, x+xi, y-yi, PSET, col, L);
        line(x-ox, y+oy, x-xi, y+yi, PSET, col, L);
        line(x-ox, y-oy, x-xi, y-yi, PSET, col, L);
        ox = xi;  oy = yi;
    }
}
chr(pat)
long pat;
{
        char *p;

        p = calloc(sizeof(char), 3);
        *p = pat >> 16;
        *(p+1) = pat >> 8;
        *(p+2) = pat;
        return(p);
}

read_cons()
{
        usrbuf[2] = 0xa;
        g_rcb.nofbyte = 3;
        g_rcb.r_byte = 14;
        subin();

        cons.cur_x = usrbuf[3];
        cons.cur_y = usrbuf[4];
        cons.atri  = usrbuf[5];
        cons.nc    = usrbuf[7];
        cons.nl    = usrbuf[8];
        cons.sl    = usrbuf[9];
        cons.ns    = usrbuf[10];
        cons.pf    = usrbuf[11];
        cons.cf    = (usrbuf[12]<<=8) + usrbuf[13];
}

read_vram()
{

        usrbuf[2] = 0x24;
        g_rcb.nofbyte = 3;
        g_rcb.r_byte = 16;
        subin();

        vram.hard = usrbuf[3];
        vram.act = usrbuf[4];
        vram.disp = usrbuf[5];
        vram.page0 = ctoi((unsigned)usrbuf[8])+usrbuf[9];
        vram.page1 = ctoi((unsigned)usrbuf[10])+usrbuf[11];
        vram.page2 = ctoi((unsigned)usrbuf[12])+usrbuf[13];
        vram.page3 = ctoi((unsigned)usrbuf[14])+usrbuf[15];
}

defpf(n, s)
int  n;
char *s;
{
        int  i = 0;
        usrbuf[2] = 0x2a;             /*  define pf string command */
        usrbuf[3] = n;                /*  pf-key number  */
        usrbuf[4] = strlen(s);        /*  string length set */
        while( *s )
             usrbuf[5+i++] = *s++;    /*  set string to usrbuf  */
```

```
	g_rcb.nofbyte = ( usrbuf[4] > 15 )?  20 : 5+usrbuf[4];
	subout();
}

getpf(n, buf)
int n;
char *buf;
{
	int  i;

	usrbuf[2] = 0x2b;		/*  get pf string command  */
	usrbuf[3] = n;			/*  pf_key number   */
	g_rcb.nofbyte = 4;
	g_rcb.r_byte  = 20;
	subin();

	for (i=0;  i<usrbuf[4]; ++i)
	    *buf++ = usrbuf[5+i];
	*buf = '¥0';
}

inkey(mode)
int  mode;
{
	 unsigned  flag;
	 int  temp;

	 switch( mode )
	 {
		case(0):			/* not wait key in */
			sr.ax = 0x0100;		/* register interface no */
			flag = sysint(222, &sr, &rr);   /* 8088 flag value */
			return((flag & 0x0001 == 1)?  -1 : rr.dx & 0x00ff);
		case(1):			/* wait from key in */
			resetkb();		 /* prohibit key interapt */
			usrbuf[2] = 0x29;
			usrbuf[3] = 1;		   /* wait! */
			g_rcb.nofbyte = 4;
			g_rcb.r_byte = 5;
			subin();
			temp = ( usrbuf[4] == 1 )?  usrbuf[3] : -1;
			setkb();		   /* enable key attention */
			return(temp);
		default:
			return(-1);
	 }
}

click(n)
int n;
{
	 sr.ax = 0x1000;
	 sr.dx = n;
	 sysint(222, &sr, &rr);
}

resetkb()
{
	 usrbuf[2] = 0x0e;
	 usrbuf[3] = 12;
	 g_rcb.nofbyte = 4;
	 subout();
}
setkb()
{
	 usrbuf[2] = 0x0f;
	 usrbuf[3] = 12;
	 g_rcb.nofbyte = 4;
	 subout();
}

getbits(x, p, n)
unsigned x, p, n;
{
	return((x >> (p+1-n)) & ~(~0 << n));
}

subout()
{
	g_rcb.reqno = 0x10;
	g_rcb.usroff = usrbuf;
	g_rcb.usrseg = usrseg();

	rcbint();  /*  call sysint function  */
}

subin()
{
```

```
     g_rcb.reqno = 0x11;
     g_rcb.usroff = usrbuf;
     g_rcb.usrseg = usrseg();

     rcbint();   /*  call sysint function  */
}

struct  {  int  csv, ssv, dsv, esv; } rrv;

usrseg()
{
     segread(&rrv);
     return(rrv.dsv);
}
     rcbint()
     {
          sr.dx = &g_rcb;
          sr.ds = g_rcb.usrseg;
          sysint(221, &sr, &rr);
     }

     gethigh(x)
     unsigned  x;
     {
          return( x >>= 8);
     }

     ctoi(c)
     unsigned  c;
     {
          return( c <<= 8);
     }
```

GEDIT　グラフィック・ツール　メイン

```
/* graphic tool program */
#include "stdio.h"
#include "stdgra.h"
#define MAX_X   639
#define MAX_Y   199
#define Y_DOT   8
#define MAXCOM  10
#define ERORR   -1
#define PROGRAM 0x100
#define SPMOV   0x200
#define NEWFILE 0
#define EXIST   2
#define SAMPLE  259                     /* sample tile string number */
#define MAXOP   8                    /* max operand number */
#define NLEN    5                    /* numerical operand max length */

/* code macro definitions  */
#define LINE    0x300
#define BOX     0x301
#define BOXFILL 0x302
#define CANCEL  0x003
#define MOVE    0x204
#define COLOR   0x30C
#define DELETE  0x20B
#define PAINT   0x305
#define TILE    0x306
#define END     0x20A
#define TRACE   0x209
#define POINT   0x307
#define GCURSOR 0x008
#define CLS     0x30D
#define INSERT  0x00F
#define ESC     0x210
#define TBEGIN  0x011
#define TEND    0x012
#define QUIT    0x013
#define CIRCLE  0x314

static int  x1, y1, x2, y2;

static int  fcolor, bcolor, ccol, hcol;         /* current color aria */

static struct prog2                     /* command save structure */
     {  int code2;   char *para;  };  /* code and parameter pointer*/
static struct prog2 *pc[4000];          /* poiner for p_code */
static int  count;                      /* program counter */
static int  cnt;                      /* current program counter */

/* parameter structure for each function */
static struct l
     { int lx1, ly1, lx2, ly2;     unsigned char lcolor, fig; };
```

グラフィック・ツール メイン

```
static struct p
     { int px, py;    unsigned char pcolor, pbnom, pbcolor[7]; };
static struct tp
     { int tx, ty;  unsigned char t1[3], t2[3], tbnom, tbcolor[7]; };
static struct ps
     { int psx, psy;      unsigned char pscolor;  };
static struct c
     { unsigned char cfcolor, cbcolor;  };
static struct cr
     { int crx, cry, rad;   unsigned char crcolor;  };

main(argc, argv)
int  argc;
char *argv[];
{
    int  flag, code, op[MAXOP];
    char command[MAXCOM];
    FILE *fopen(), *fp;

    if ( argc != 2 )
         abort("-file name is required¥n");
    fp = fileopen(*++argv, &flag);
    if ( fp == NULL )
         abort("file erorr has ocurred¥n");
    else
    {
         screen(0,0,0);                  /* screen mode select */
         cls(0);
         read_cl();                        /* color pattern reading */
         if ( flag == EXIST )
         {
              readprog(fp);
              trace(0, count);
              x1 = x2;    y1 = y2;
         }
         else                            /* newly made file */
         {
               /* intializing */
              x1 = MAX_X/2;    y1 = MAX_Y/2;
              count = 0;
              fcolor = 7;   bcolor = 0;
         }
    }
    hcol = 5;                          /* command line char color */
    ccol = 6;                          /* gcursor default color */
    cnt = count;                      /* set ep to cp  */

    cls(1);
    while( code != END  && code != QUIT )
    {
        pushfd();
        header(*argv);
        locate(14,0);
        color(7,0);
        flag = getcom(command, op);               /* get command */
        popfd();
        if ( flag != ERORR && (code=c_code2(command)) != ERORR¥
             && optest(0,code,flag,op) != ERORR )
             c_exec2(code, flag, op, *argv);    /* command exec */
        else
              beep();                    /* command or parameter erorr */
        cls(1);
    }
    trace(cnt, count);
    cnt = count;
    if ( code == END )
        edit_end(*argv, fp);
}

dinput(endcode, convert, filename)
int  endcode, (*convert)();
char *filename;
{
    char command[MAXCOM];
    unsigned char tile1[3], tile2[3];
    int  flag, op[MAXOP], code, ecode;
    struct prog2 *wp;

    code |= SPMOV;
    wp = calloc(1, sizeof(struct prog2));
    while( code != endcode )
    {
        g_cursor(ccol);
        if ((code & SPMOV) != SPMOV)       /* not move start point */
             popsp(&x1, &y1);
        do
        {
          pushfd();
```

```
            cls(1);
            header(filename);
            locate(14,0);      color(7,0);
            flag = getcom(command,op);          /* get command 1 */
            if (flag != ERORR && (code = (*convert)(command)) != ERORR)
            {
                if ( code == TILE )
                    tileget(tile1, tile2);
                popfd();          /* pop header field */
                if ((ecode = optest(wp, code,  flag, op, tile1, tile2))¥
                    != ERORR )
                {
                    if ((code & SPMOV) == SPMOV)
                    {
                        wp->code2 = code;
                        popfd();
                        era_sp(x1, y1);
                        if ((code & PROGRAM) == PROGRAM)
                        {
                            c_exec1(wp);
                            progsave(wp);
                        }
                        else
                            c_exec2(code, flag, op, filename);
                        disp_sp(x2, y2);
                    }
                    else
                    {
                        pushsp(x1, y1);
                        if ( code == GCURSOR )
                            ccol = op[0];
                    }
                }
            }
            else
            {
                popfd();
                ecode = ERORR;
            }
            if ( ecode == ERORR )
                beep();
        }while ( ecode == ERORR );
        cls(1);
        x1 = x2;   y1 = y2;
    }
    free(wp);
}


insert(filename)
char *filename;
{
     int  c_code();

     hcol = 4;
     cls(1);
     disp_sp(x1, y1);
     dinput(ESC, c_code, filename);
     hcol = 5;
     era_sp(x1, y1);
}

trc_exec(flag, op)
int  flag, *op;
{
     int  from, before;

     *op = ( flag == 0 )?  1 : *op;
     if ( *op > 0 )
     {
           from = cnt;
           cnt = before = ( cnt+*op > count )?  count : cnt+*op;
     }
     else
     {
           from = 0;                   /* from beginning */
           cnt = before = ( cnt+*op < 0 )?  0 : cnt+*op;
     }
     trace(from, before);
     x1 = x2;          y1 = y2;        /* set current position */
}

progsave(p)
struct prog2 *p;
{
      int  i;

      for (i=count; i>cnt; --i)
```

グラフィック・ツール　メイン

```
            pc[i] = pc[i-1];
        pc[cnt] = (struct prog2 *)calloc(1, sizeof(struct prog2));
        movmem(p,pc[cnt], sizeof(struct prog2));
        count++;
        cnt++;
}

pushfd()
{
        saveg(0x0000, 0x0000, 80, Y_DOT, 0x07, 0x17);
}


popfd()
{
        saveg(0x0000, 0x0000, 80, Y_DOT, 0x17, 0x07);
}

header(filename)
char *filename;
{
        line(0, 0, 639, Y_DOT-1, PSET, 0, BF);
        color(hcol,0);
        locate(0,0);            printf("%s", filename);
        locate(32,0);           printf("CP=%04d EP=%04d", cnt, count);
        locate(9,0);
        printf("CMD?=");
        locate(53,0);
        printf("X1=%3d Y1=%3d X2=%3d Y2=%3d", x1, y1, x2, y2);
        line(383,0,416,Y_DOT-1,PSET,7,B);
        line(384,1,399,Y_DOT-2,PSET,fcolor,BF);     /* display fcolor */
        line(400,1,415,Y_DOT-2,PSET,bcolor,BF);     /* display bcolor */
        color(fcolor, bcolor);
}

int  xbuf, ybuf;

pushsp(x, y)
int x, y;
{
        xbuf = x;
        ybuf = y;
}
popsp(x, y)
int  *x, *y;
{
        *x = xbuf;
        *y = ybuf;
}

c_exec1(p)
struct prog2 *p;
{
          switch(p->code2)
          {
                case( LINE ):
                case( BOX  ):
                case( BOXFILL ):
                        lineexec((struct l *)(p->para));
                        break;
                case( POINT ):
                        psexec((struct ps *)(p->para));
                        break;
                case( PAINT ):
                        ptexec((struct p *)(p->para));
                        break;
                case( TILE ):
                        tileexec((struct tp *)(p->para));
                        break;
                case( CIRCLE ):
                        crexec((struct cr *)(p->para));
                        break;
                case( COLOR ):
                        colexec((struct c *)(p->para));
                        break;
                case( CLS ):
                        cls(5);         /* graphic screen clear */
                        break;
          }
}

c_exec2(code, flag, op, filename)
int  code, flag, *op;
char *filename;
{
        switch( code )
        {
            case( MOVE ):
```

```
				break;
			case( DELETE ):
				delete(flag, op);
				break;
			case( QUIT ):
			case( END ):
				break;
			case( TRACE ):
				trc_exec(flag, op);
				break;
			case( INSERT ):
				insert(filename);
				break;
			case( TBEGIN ):
				cnt = 0;
				trace(0,cnt);
				break;
			case( TEND ):
				trace(cnt, count);
				cnt = count;
				break;
			default:
				break;
		}
		return;
}

optest(p, code, flag, op, t1, t2)
int  code, flag, *op;
unsigned char *t1, *t2;
struct prog2  *p;
{
	int  ecode;

	switch( code )
	{
		case( LINE ):
			ecode = linetest(p,flag,op,x1, y1, x2, y2, fcolor, L);
			break;
		case( BOX ):
			ecode = linetest(p,flag,op,x1, y1, x2, y2, fcolor, B);
			break;
		case( BOXFILL ):
			ecode = linetest(p,flag,op,x1,y1, x2, y2, fcolor, BF);
			break;
		case( POINT ):
			ecode = pointest(p, flag, op, x2, y2, fcolor);
			break;
		case( PAINT ):
			ecode = pttest(p, flag, op, x2, y2, bcolor);
			break;
		case( TILE ):
			ecode = tiletest(p, flag, op, x2, y2, bcolor, t1, t2);
			break;
		case( COLOR ):
			ecode = coltest(p, flag, op);
			break;
		case( CIRCLE ):
			ecode = crtest(p, flag, op, x2, y2, fcolor);
			break;
		case( CLS ):
			ecode = clstest(p, flag);
			break;
		case( TBEGIN ):
		case( TEND ):
		case( INSERT ):
		case( CANCEL ):
		case( MOVE ):
		case( QUIT ):
		case( END ):
			ecode = nooptest(flag);
			break;
		case( DELETE ):
			ecode = deltest(flag, op);
			break;
		case( TRACE ):
			ecode = trctest(flag);
			break;
		case( GCURSOR ):
			ecode = gctest(flag, op);
			break;
		case( ESC ):
			ecode = 0;
			break;
		default:
			return(-1);
	}
	return(ecode);
}
```

グラフィック・ツール　メイン

```
fileopen(filename, f)
char *filename;
int  *f;
{
      FILE *fopen(), *fp;
      char s[11];

      if ( strlen(filename)> 8 )
            return(NULL);
      strcpy(s, filename);
      strcat(s, ".g");
      fp = fopen(s, "rwb");
      if ( fp == NULL )
      {
          fp = fopen(s, "wb");
          *f = NEWFILE;
      }
      else
          *f = EXIST;
      return(fp);
}

edit_end(filename, fp)
char *filename;
FILE *fp;
{
      char ps[100], s[11];
      int  i;
      FILE *fopen();

      wtprog(fp);                     /*  struct prog output */
      fclose(fp);

      strcpy(s, filename);
      strcat(s, ".c");
      fp = fopen(s, "w");
      fprintf(fp, "#include %cstdio.h%c¥n#include %cstdgra.h%c¥n",¥
       0x22, 0x22, 0x22, 0x22);
      fprintf(fp, "main()¥n{¥n");
      locate(0,1);
      for(i=0; i<count; ++i)
            if (makeprog(ps, i) != -1)
            {
                  printf(ps);
                  fprintf(fp, "     %s", ps);
            }
      fprintf(fp, "}¥n");
      fclose(fp);
}

readprog(fp)
FILE *fp;
{
      int  i, size;

      fread(&count, sizeof(int), 1, fp);

      for (i=0; i<count; ++i)
      {
          pc[i] = (struct prog2 *)calloc(1, sizeof(struct prog2));
          fread(&pc[i]->code2, sizeof(int), 1, fp); /* code read */
          size = p_size(pc[i]->code2);
          pc[i]->para = calloc(1, size);
          fread(pc[i]->para, size, 1, fp);
      }
      fseek(fp, 0L, 0);               /*  rewind head of file */
}

wtprog(fp)
FILE *fp;
{
      int  i;
      fwrite(&count, sizeof(int), 1, fp);

      for (i=0; i<count; ++i)
      {
            fwrite(&pc[i]->code2, sizeof(int), 1, fp);
            fwrite(pc[i]->para, p_size(pc[i]->code2), 1, fp);
      }
}

p_size(code)
int  code;
{
      int  size;

      switch( code )
      {
```

```
		case( LINE ):
		case( BOX ):
		case( BOXFILL ):
			size = sizeof(struct l);
			break;
		case( POINT ):
			size = sizeof(struct ps);
			break;
		case( PAINT ):
			size = sizeof(struct p);
			break;
		case( TILE ):
			size = sizeof(struct tp);
			break;
		case( COLOR ):
			size = sizeof(struct c);
			break;
		case( CIRCLE ):
			size = sizeof(struct cr);
			break;
		case( CLS ):
			size = 0;		/* no parameter is required */
			break;
		default:
			size = -1;
			break;
	}
	return(size);
}

gctest(flag, op)
int	flag, *op;
{
	return((flag == 1 && *op >= 0 && *op <= 7)?  0 : -1);
}

deltest(flag, op)
int	flag, *op;
{
	return( ((flag==1 && *op>=0 || flag==0) && cnt>0)?  0 : -1);
}

delete(flag, op)
int	flag, *op;

{
	int	dn, i;

	if ( flag == 0 )
		dn = 1;
	else
		dn = ( *op > cnt )?  cnt : *op;

	for (i=cnt-dn;  i<cnt; ++i)
	{
		if (pc[i]->para != NULL)
			free(pc[i]->para);
		free(pc[i]);
	}
	for (i=cnt; i<count; ++i)
		pc[i-dn] = pc[i];			/* slide array */
	count -= dn;
	cnt -= dn;
	trace(0, cnt);
	x1 = x2;	y1 = y2;
}

makeprog(s, n)
char *s;
int  n;
{
	int	i, j, cd, ecode;

	cd = pc[n]->code2;
	if ( cd == LINE || cd == BOX || cd == BOXFILL )
	{
		struct l *p;
		p = pc[n]->para;
		sprintf(s, "line(%d, %d, %d, %d, PSET, %d, %d);¥n",¥
		 p->lx1, p->ly1, p->lx2, p->ly2, p->lcolor, p->fig);
		ecode = 0;
	}
	else if ( cd == PAINT )
	{
		struct p *p;
		p = pc[n]->para;
		sprintf(s, "paint(%d, %d, %d, %d", p->px, p->py, ¥
		  p->pcolor, p->pbnom);
```

グラフィック・ツール　メイン

```
            for (j=0; j<p->pbnom; ++j)
            {
                s += strlen(s);
                sprintf(s, ", %d", p->pbcolor[j]);
            }
            sprintf( s += strlen(s), ");\n");
            ecode = 0;
        }
        else if ( cd == TILE )
        {
            struct tp *p;
            p = pc[n]->para;
            sprintf(s,"tpaint(%d, %d, 2, chr(0x%02x%02x%02xL), chr(0x%02x%02x%02xL), 0, 0, %d",\
             p->tx, p->ty, p->t1[0], p->t1[1], p->t1[2], p->t2[0], \
             p->t2[1], p->t2[2], p->tbnom);
            for (j=0; j<p->tbnom; ++j)
            {
                s += strlen(s);
                sprintf(s, ", %d", p->tbcolor[j]);
            }
            sprintf( s += strlen(s), ");\n");
            ecode = 0;
        }
        else if ( cd == POINT )
        {
            struct ps *p;
            p = pc[n]->para;
            sprintf( s, "pset(%d, %d, %d, PSET);\n", p->psx, p->psy,\
             p->pscolor);
             ecode = 0;
        }
        else if ( cd == COLOR )
        {
          struct c *p;      p = pc[n]->para;
          sprintf(s, "color(%d, %d);\n", p->cfcolor, p->cbcolor);
          ecode = 0;
        }
        else if ( cd == CIRCLE )
        {
          struct cr *p;      p = pc[n]->para;
          sprintf(s, "circle(%d, %d, %d, %d);\n", p->crx, p->cry,\
             p->rad, p->crcolor);
          ecode = 0;
        }
        else if ( cd == CLS )
        {
            sprintf(s, "cls(5);\n");
            ecode = 0;
        }
        else
            ecode = -1;
        return(ecode);
}

nooptest(flag)
int  flag;
{
      return( (flag == 0)?  0 : -1 );
}

tiletest(p, flag, op, x, y, bc, t1, t2)
struct prog2 *p;
int  flag, *op, x, y;
unsigned char *t1, *t2, bc;
{
      struct tp *tpp;
      int  i, j, ecode;

      for (i=0; i<MAXOP && bittest(flag,i) == 1;  ++i)
          ;
      if ( i == 0 || i >= MAXOP-1 )
          ecode = 0;
      else
      {
          for (j=i+1; j<MAXOP && bittest(flag, i) == 0; ++j)
              ;
          if ( j != MAXOP )
              ecode = -1;
          else
              ecode = 0;
      }

      if ( !ecode )                  /* operand number check is ok */
          for (j=0; j<i; ++j)             /* color code check */
              if ( *(op+j)<0 || *(op+j)>7 )
                  ecode = -1;
      if ( !ecode )
      {
```

```
        tpp = calloc(1, sizeof(struct tp));    /* allocate struct tp */
        if ( i == 0 )       /* when border color is omitted */
        {
            for (j=0,i=0; j<8; ++j) /*default border color set*/
                if ( j != bc )     /* except background color */
                    tpp->tbcolor[i++] = j;
            i = 7;              /*  7 colors is border */
        }
        else
             for (j=0; j<i; ++j)
                  tpp->tbcolor[j] = *(op+j);
        tpp->tx = x;         tpp->ty = y;
        movmem(t1, tpp->t1, 3);    movmem(t2, tpp->t2, 3);
        tpp->tbnom = i;
        p->para = tpp;          /* tell struct prog2 where para is */
    }
    return( ecode );
}

tileexec(p)
struct tp *p;
{
    tpaint((x2=p->tx), (y2=p->ty), 2, p->t1, p->t2, 0, 0, p->tbnom, ¥
       p->tbcolor[0], p->tbcolor[1], p->tbcolor[2], p->tbcolor[3],¥
       p->tbcolor[4], p->tbcolor[5], p->tbcolor[6]);
}

pttest(p, flag, op, x, y, bc)
struct prog2 *p;
int  flag, *op, x, y;
unsigned char bc;
{
    struct p *pp;
    int  i, j, ecode;

    for (i=0; i<MAXOP && bittest(flag,i) == 1;  ++i)
          ;
    if ( i == 0 )                    /* no operand */
         ecode = -1;
    else if ( i >= MAXOP-1 )
         ecode = 0;
    else
    {
         for (j=i+1; j<MAXOP && bittest(flag, i) == 0; ++j)
              ;
         if ( j != MAXOP )
              ecode = -1;
         else
              ecode = 0;

    }

    if ( !ecode )                   /* operand number check is ok */
        for (j=0; j<i; ++j)              /* color code check */
            if ( *(op+j)<0 || *(op+j)>7 )
                 ecode = -1;
    if ( !ecode )
    {
        pp = calloc(1, sizeof(struct p));
        if ( i == 1 )        /* when border color is omitted */
        {
            for (j=0,i=0; j<8; ++j) /*default border color set*/
                if ( j != bc )     /* except background color */
                    pp->pbcolor[i++] = j;
            i = 8;
        }
        else
            for (j=0; j<i-1; ++j)
                 pp->pbcolor[j] = *(op+j+1);
        pp->px = x;          pp->py = y;
        pp->pcolor = *op;
        pp->pbnom = i - 1;
        p->para = pp;
    }
    return( ecode );
}

ptexec(p)
struct p *p;
{
    paint((x2=p->px), (y2=p->py), fcolor=p->pcolor, p->pbnom,¥
      p->pbcolor[0], p->pbcolor[1], p->pbcolor[2], p->pbcolor[3],¥
      p->pbcolor[4], p->pbcolor[5], p->pbcolor[6]);
}

trctest(flag)
int  flag;
{
```

グラフィック・ツール　メイン

```
	return( (flag == 0 || flag == 1)? 0 : -1 );
}

trace(from, before)
int  from, before;
{
	char  s[100];
	int  i;
	if ( from == 0 )
	{
		color((fcolor=7), (bcolor=0));
		x2 = x1 = MAX_X/2;   y2 = y1 = MAX_Y/2;
		cls(5);
	}

	for (i=from; i<before; ++i)
	    c_exec1(pc[i]);
	x1 = x2;        y1 = y2;          /* set current position */
	return(0);
}

pointest(p, flag, op, x, y, fc)
struct prog2 *p;
int  flag, *op, x, y;
unsigned char fc;
{
      struct ps *psp;

      if ( flag != 0 && flag != 1 )
            return(-1);
      else if ( flag == 1 && ( *op<0 || *op>7) )
            return(-1);
      else
      {
          psp = calloc(1, sizeof(struct ps));
          psp->psx = x;    psp->psy = y;
          psp->pscolor = (flag == 1)?  *op : fc;
          p->para = psp;
          return(0);
      }
}


coltest(p, flag, op)
struct prog2 *p;
int  flag, *op;
{
    struct c *cp;

    if ( flag > 3 )
        return(-1);            /* ERORR */
    else
    {
        cp = (struct c *)calloc(1, sizeof(struct c));
        cp->cfcolor = (bittest(flag, 0) == 1)?  *op : fcolor;
        cp->cbcolor = (bittest(flag, 1) == 1)?  *(op+1) : bcolor;
        p->para = cp;
        return(0);
    }
}

colexec(p)
struct c *p;
{
	color((fcolor=p->cfcolor), (bcolor=p->cbcolor));
}

clstest(p, flag)
struct prog2 *p;
int  flag;
{
      if ( flag != 0 )
          return(-1);
      else
      {
          p->para = NULL;          /* no parameter */
          return(0);
      }
}

crtest(p, flag, op, x2, y2, fc)
int  flag, *op, x2, y2;
unsigned char fc;
struct prog2 *p;
{
    struct cr *crp;
    if ( flag != 1 && flag != 3 )
          return(-1);
```

```
    else if ( flag == 3 && ( *(op+1)<0 || *(op+1)>7 ))
        return(-1);
    else
    {
        crp = calloc(1, sizeof(struct cr));
        crp->crx = x2;          crp->cry = y2;
        crp->rad = *op;
        crp->crcolor = ( flag == 3 )?   *(op+1) : fc;
        p->para = crp;
        return(0);
    }
}

crexec(p)
struct cr *p;
{
      circle( (x2=p->crx), (y2=p->cry), p->rad, fcolor=p->crcolor);
}

linetest(p, flag, op, x1, y1, x2, y2, fc, fig)
int  flag, *op, x1, y1, x2, y2;
unsigned char fig, fc;
struct prog2 *p;
{
     struct l *lp;
     if ( flag != 0 && flag != 1 )
          return(-1);
     else if ( flag == 1 && ( *op<0 || *op>7) )
          return(-1);
     else
     {
           lp = calloc(1, sizeof(struct l));
           lp->lx1 = x1;          lp->ly1 = y1;
           lp->lx2 = x2;          lp->ly2 = y2;
           lp->lcolor = (flag == 1)?  *op : fc;
           lp->fig = fig;
           p->para = lp;
           return(0);                      /* non error */
     }
}

lineexec(p)
struct l *p;
{
       line(p->lx1, p->ly1, (x2=p->lx2), (y2=p->ly2), PSET,¥
          fcolor=p->lcolor, p->fig);
}

psexec(p)
struct ps *p;
{
     pset((x2=p->psx), (y2=p->psy), fcolor=p->pscolor, PSET);
}

getcom(com, op)
char *com;
int  *op;
{
     char  ws[NLEN];
     int  c, flag, n[MAXOP], i, count;

     for (i=0; i<MAXOP; ++i)
         n[i] = 0;
     flag &= 0;

     while((c=getchar()) == ' ' || c == '¥n' || c == '¥t')
          ;                              /* skip space */
     if (( c < 'a' || c > 'z') && (c <'A' || c > 'Z') && c != 0x1b)
     {
           clr_buf();
           return(-1);
     }
     else
           *com++ = tolower(c);
     while((c=getchar())>='a' && c<='z' || c>='A' && c<='Z')
           *com++ = tolower(c);
     *com = '¥0';
     if ( c == '¥n' )
           return(flag);
     else
     {
          ungetc(c, stdin);
          i = 0;
          count = 0;
          do
          {
               c = getchar();
               if ((c=='-' || c=='+' || c>='0' && c<='9') && i<NLEN-1)
```

```
                ws[i++] = c;
            else if ( c == ',' || c == '¥n' )
            {
                if ( i == 0 )
                {
                    n[count] = 0;
                    flag &= ~power(2,count);
                }
                else
                {
                    ws[i] = '¥0';
                    n[count] = atoi(ws);
                    flag |= power(2, count);
                }
                count++;
                i = 0;
            }
        } while( c != '¥n' && count < MAXOP);
        if ( c != '¥n' )
            clr_buf();                      /* clear buffer */
        for (i=0; i<MAXOP; ++i)
            *op++ = n[i];
        return(flag);
    }
}

clr_buf()
{
        while( getchar() != '¥n' )
            ;                           /*  clear buffer */
}

power(x, n)
int x, n;
{
    int p;
    for ( p=1; n>0; --n)
        p = p*x;
    return(p);
}

#define COMNUM  13

c_code(command)
char *command;
{
    static char *c_name[] =
    {
        "l", "b", "bf", "can", "c", "p", "tp", ¥
         "po", "gc", "m", "cls", "¥033", "cr"
    };
    static int c_tab[] =
    {
        LINE, BOX, BOXFILL, CANCEL, COLOR, PAINT, TILE,¥
        POINT, GCURSOR, MOVE, CLS, ESC, CIRCLE
    };
    int  i;

    for (i=0; i<COMNUM; i++)
        if ( strcmp(command, c_name[i]) == 0 )
                return(c_tab[i]);
    return(-1);
}

#define COMNUM2  7

c_code2(command)
char *command;
{
    static char *c_name2[] =
    {
        "t", "d", "i", "e", "tb", "te", "quit"
    };
    static int c_tab2[] =
    {
        TRACE, DELETE, INSERT, END, TBEGIN, TEND, QUIT
    };
    int  i;

    for (i=0; i<COMNUM2; i++)
        if ( strcmp(command, c_name2[i]) == 0 )
                return(c_tab2[i]);
    return(-1);
}
bittest(x, n)
int x, n;
{
     return(( x>>n ) & 0x0001 );
}
```

```
extern unsigned char  usrbuf[];

extern struct rcb
      {
            unsigned char  reqno, status;
            unsigned  usroff, usrseg;
            unsigned char  nofbyte, null, reserve, r_byte;
      } g_rcb;

g_cursor(color)
int  color;
{
      int  i;
      resetkb();
      usrbuf[2] = 0x1f;
      usrbuf[3] = color;
      usrbuf[4] = 1;
      usrbuf[5] = gethigh(x1);    usrbuf[6] = x1;
      usrbuf[7] = gethigh(y1);    usrbuf[8] = y1;
      g_rcb.nofbyte = 9;
      g_rcb.r_byte = 7;
      subin();
      x2 = ctoi((unsigned)usrbuf[3]) + usrbuf[4];
      y2 = ctoi((unsigned)usrbuf[5]) + usrbuf[6];
      setkb();
}

saveg(sa, da, byte, lines, sc, dc)
unsigned  sa, da, byte, lines;
unsigned char sc, dc;
{
      usrbuf[2] = 0x33;                /* copy graphic vram command */
      usrbuf[3] = sc;
      usrbuf[4] = gethigh(sa);    usrbuf[5] = sa;
      usrbuf[6] = gethigh(byte);   usrbuf[7] = byte;
      usrbuf[8] = gethigh(lines);  usrbuf[9] = lines;
      usrbuf[10] = dc;
      usrbuf[11] = gethigh(da);    usrbuf[12] = da;
      g_rcb.nofbyte = 13;
      subout();
}

unsigned char *spp;

disp_sp(x,y)
int  x, y;
{
      int  x1, y1, x2, y2;
      unsigned char *get_ga();

      x1 = ( x-4 >= 0 )?  x-4 : 0;
      y1 = ( y-2 >= 0 )?  y-2 : 0;
      x2 = ( x+4 <= MAX_X )?  x+4 : MAX_X;
      y2 = ( y+2 <= MAX_Y )?  y+2 : MAX_Y;
      spp = get_ga(x1, y1, x2, y2);
      line(x1, y1, x2, y2, PSET, 7, B);
}

era_sp(x, y)
int  x, y;
{
      int  x1, y1, x2, y2;
      unsigned char *get_ga();

      x1 = ( x-4 >= 0 )?  x-4 : 0;
      y1 = ( y-2 >= 0 )?  y-2 : 0;
      x2 = ( x+4 <= MAX_X )?  x+4 : MAX_X;
      y2 = ( y+2 <= MAX_Y )?  y+2 : MAX_Y;
      put_ga(x1, y1, x2, y2, spp, PSET);
      free(spp);
}

static unsigned char *tl1[SAMPLE], *tl2[SAMPLE];
#define FORWARD  1
#define BACK     0

tileget(t1, t2)
unsigned char *t1, *t2;
{
      int  n, start, mode;

      fd_save();                  /* save display field */
      frame();                          /* display frame */
      color(7,0);
      cldisp(start=0);          /*  display color 24 */
      mode = FORWARD;
      while((n=nget()) < 0 || n > SAMPLE-1 )
      {
```

グラフィック・ツール　メイン

```
		if ( n == -2 )
			mode = BACK;
		else if ( n == -3 )
			mode = FORWARD;
		if ( mode == FORWARD && (start+24)<241 )
			start += 24;
		else if ( mode == BACK && (start-24)>=0 )
			start -= 24;
		frame();
		cldisp(start);
	}
	movmem(tl1[n], t1, 3);			/* set tile string to para */
	movmem(tl2[n], t2, 3);
	fd_pop();
	cls(1);
}

nget()
{
	int  i, c;
	int  ecode = -1;
	char s[10];

	while( ecode == -1 )
	{
		i = 0;
		locate(9,0);
		printf("PATTERN?=              ");
		locate(18,0);
		while( (c=getchar()) != '¥n' )
		{
			while( c == ' ' )
				;
			if ( c == '+' || c == '-' || c>='0' && c<='9'
			{
				s[i] = c;
				ecode = 0;
			}
			else
				ecode = -1;
			i += ( i+1 < 10 )?  1 : 0;
		}
		if ( i == 0 )
				ecode = 0;
	}
	if ( i == 0 )
		return(-1);
	else if ( s[0] == '-' )
		return(-2);
	else if ( s[0] == '+' )
		return(-3);
	else
		return(atoi(s));
}

cldisp(start)
int start;
{
	int  i;

	for (i=0; i<24; ++i)
	{
		locate(75,i+1);
		if ( i+start < SAMPLE )
		{
			printf("%3d", i+start);
			tpaint(632,i*Y_DOT+Y_DOT+4,2,tl1[i+start],¥
			 tl2[i+start],0,0,1,7);
		}
		else
			printf("   ");
	}
}
 read_cl()
 {
	char *alloc();
	int  i, c, j;
	FILE  *fp, *fopen();

	fp = fopen("tcolor.dat", "rb");
	fread(&c, sizeof(int), 1, fp);

	for (i=0; i<c; ++i)
	{
		tl1[i] = alloc(3);
		tl2[i] = alloc(3);
		fread(tl1[i], sizeof(char), 3, fp);
```

```
            fread(tl2[i], sizeof(char), 3, fp);
        }
        fclose(fp);
        return(c);
}

fd_save()
{
        saveg(0x02cb, 0x02cb, 5, 192, 0x07, 0x17);
}

fd_pop()
{
        saveg(0x02cb, 0x02cb, 5, 192, 0x17, 0x07);
}

frame()
{
        int  i;

        line(600,Y_DOT,MAX_X,MAX_Y,PSET,0,BF);
        line(624,Y_DOT,MAX_X,MAX_Y,PSET,7,B);
        for (i=15; i<=MAX_Y; i+=Y_DOT)
             line(624, i, MAX_X, i, PSET, 7, L);
}
```

RUN

```
/* graphic tool program */
#include "stdio.h"
#include "stdgra.h"
#define MAX_X    639
#define MAX_Y    199
#define Y_DOT    8
#define MAXCOM   10
#define ERORR   -1
#define PROGRAM 0x100
#define SPMOV   0x200
#define NEWFILE 0
#define EXIST   2
#define SAMPLE  259                    /* sample tile string number */
#define MAXOP  8                    /* max operand number */
#define NLEN   5                    /* numerical operand max length */

/* code macro definitions  */
#define LINE    0x300
#define BOX     0x301
#define BOXFILL 0x302
#define CANCEL  0x003
#define MOVE    0x204
#define COLOR   0x30C
#define DELETE  0x20B
#define PAINT   0x305
#define TILE    0x306
#define END     0x20A
#define TRACE   0x209
#define POINT   0x307
#define GCURSOR 0x008
#define CLS     0x30D
#define INSERT  0x00F
#define ESC     0x210
#define TBEGIN  0x011
#define TEND    0x012
#define QUIT    0x013
#define CIRCLE  0x314

static int  x1, y1, x2, y2;

static int  fcolor, bcolor, ccol, hcol;        /* current color aria */

static struct prog2                       /* command save structure */
     { int code2;   char *para;  };   /* code and parameter pointer*/
static struct prog2 *pc[5000];            /* poiner for p_code */
static int  count;                        /* program counter */
static int  cnt;                         /* current program counter */

/* parameter structure for each function */
static struct l
     { int lx1, ly1, lx2, ly2;     unsigned char lcolor, fig; };
static struct p
     { int px, py;    unsigned char pcolor, pbnom, pbcolor[7]; };
static struct tp
     { int tx, ty;  unsigned char t1[3], t2[3], tbnom, tbcolor[7]; };
static struct ps
     { int psx, psy;     unsigned char pscolor;  };
static struct c
     { unsigned char cfcolor, cbcolor;  };
```

RUN

```
static struct cr
    { int crx, cry, rad;  unsigned char crcolor;  };

main(argc, argv)
int  argc;
char *argv[];
{
    int  flag, code, op[MAXOP];
    char command[MAXCOM];
    FILE *fopen(), *fp;

    if ( argc != 2 )
        abort("-file name is required¥n");
    fp = fileopen(*++argv, &flag);
    if ( fp == NULL )
        abort("file erorr has ocurred¥n");
    else
    {
        screen(0,0,0);                  /* screen mode select */
        cls(0);
        if ( flag == EXIST )
        {
            readprog(fp);
            trace(0, count);
        }
    }
}

trc_exec(flag, op)
int  flag, *op;
{
     int  from, before;

     *op = ( flag == 0 )?  1 : *op;
     if ( *op > 0 )
     {
          from = cnt;
          cnt = before = ( cnt+*op > count )?  count : cnt+*op;
     }
     else
     {
          from = 0;                      /* from beginning */
          cnt = before = ( cnt+*op < 0 )?  0 : cnt+*op;
     }
     trace(from, before);
     x1 = x2;          y1 = y2;          /* set current position */
}


c_exec1(p)
struct prog2 *p;
{
       switch(p->code2)
       {
             case( LINE ):
             case( BOX  ):
             case( BOXFILL ):
                   lineexec((struct l *)(p->para));
                   break;
             case( POINT ):
                   psexec((struct ps *)(p->para));
                   break;
             case( PAINT ):
                   ptexec((struct p *)(p->para));
                   break;
             case( TILE ):
                   tileexec((struct tp *)(p->para));
                   break;
             case( CIRCLE ):
                   crexec((struct cr *)(p->para));
                   break;
             case( COLOR ):
                   colexec((struct c *)(p->para));
                   break;
             case( CLS ):
                   cls(5);          /* graphic screen clear */
                   break;
       }
}


fileopen(filename, f)
char *filename;
int  *f;
{
     FILE *fopen(), *fp;
     char s[11];
```

```
	if ( strlen(filename)> 8 )
		return(NULL);
	strcpy(s, filename);
	strcat(s, ".g");
	fp = fopen(s, "rwb");
	if ( fp == NULL )
	{
		fp = fopen(s, "wb");
		*f = NEWFILE;
	}
	else
		*f = EXIST;
	return(fp);
}


readprog(fp)
FILE *fp;
{
	int  i, size;

	fread(&count, sizeof(int), 1, fp);

	for (i=0; i<count; ++i)
	{
		pc[i] = (struct prog2 *)calloc(1, sizeof(struct prog2));
		fread(&pc[i]->code2, sizeof(int), 1, fp); /* code read */
		size = p_size(pc[i]->code2);
		pc[i]->para = calloc(1, size);
		fread(pc[i]->para, size, 1, fp);
	}
	fseek(fp, 0L, 0);			/*  rewind head of file */
}


p_size(code)
int  code;
{
	int  size;

	switch( code )
	{
		case( LINE ):
		case( BOX ):
		case( BOXFILL ):
			size = sizeof(struct l);
			break;
		case( POINT ):
			size = sizeof(struct ps);
			break;
		case( PAINT ):
			size = sizeof(struct p);
			break;
		case( TILE ):
			size = sizeof(struct tp);
			break;
		case( CIRCLE ):
			size = sizeof(struct cr);
			break;
		case( COLOR ):
			size = sizeof(struct c);
			break;
		case( CLS ):
			size = 0;		/* no parameter is required */
			break;
		default:
			size = -1;
			break;
	}
	return(size);
}

tileexec(p)
struct tp *p;
{
	tpaint((x2=p->tx), (y2=p->ty), 2, p->t1, p->t2, 0, 0, p->tbnom, ¥
	  p->tbcolor[0], p->tbcolor[1], p->tbcolor[2], p->tbcolor[3],¥
	  p->tbcolor[4], p->tbcolor[5], p->tbcolor[6]);
}


ptexec(p)
struct p *p;
{
	paint((x2=p->px), (y2=p->py), fcolor=p->pcolor, p->pbnom,¥
	  p->pbcolor[0], p->pbcolor[1], p->pbcolor[2], p->pbcolor[3],¥
	  p->pbcolor[4], p->pbcolor[5], p->pbcolor[6]);
}
```

RUN

```
trace(from, before)
int  from, before;
{
      char  s[100];
      int  i;
      if ( from == 0 )
      {
            color((fcolor=7), (bcolor=0));
            x2 = x1 = MAX_X/2;   y2 = y1 = MAX_Y/2;
            cls(5);
      }

      for (i=from; i<before; ++i)
          c_exec1(pc[i]);
      x1 = x2;        y1 = y2;            /* set current position */
      return(0);
}

crtest(p, flag, op, x2, y2, fc)
int  flag, *op, x2, y2;
unsigned char fc;
struct prog2 *p;
{
    struct cr *crp;
    if ( flag != 1 && flag != 3 )
         return(-1);
    else if ( flag == 3 && ( *(op+1)<0 || *(op+1)>7 ))
         return(-1);
    else
    {
         crp = calloc(1, sizeof(struct cr));
         crp->crx = x2;         crp->cry = y2;
         crp->rad = *op;
         crp->crcolor = ( flag == 3 )?   *(op+1) : fc;
         p->para = crp;
         return(0);
    }
}

crexec(p)
struct cr *p;
{
      circle( (x2=p->crx), (y2=p->cry), p->rad, fcolor=p->crcolor);
}



colexec(p)
struct c *p;
{
      color((fcolor=p->cfcolor), (bcolor=p->cbcolor));
}

clstest(p, flag)
struct prog2 *p;
int  flag;
{
     if ( flag != 0 )
          return(-1);
     else
     {
          p->para = NULL;          /* no parameter */
          return(0);
     }
}

lineexec(p)
struct l *p;
{
       line(p->lx1, p->ly1, (x2=p->lx2), (y2=p->ly2), PSET,\
          fcolor=p->lcolor, p->fig);
}

psexec(p)
struct ps *p;
{
     pset((x2=p->psx), (y2=p->psy), fcolor=p->pscolor, PSET);
}


power(x, n)
int x, n;
{
    int p;
    for ( p=1; n>0; --n)
         p = p*x;
    return(p);
}
```

FM-7/11マシン語ファイルやデータ・ファイルを簡単にコピー！

FileCopy PROGRAM

ファイルコピープログラム

■中安博司

最近フロッピーディスク装置が求めやすくなり，読者の中にもディスクを使う人が増えてきていると思います．確かにディスクは便利なのですが，枚数が増えてくるとその整理はけっこう大変なものです．ファイルを他のディスケットに移し変えようとしたとき，BASICプログラムの場合には，LOAD，SAVEの繰り返しさえいとわなければ何とかできますが，マシン語プログラムは開始番地などをいちいち調べる必要があり，データ・ファイルのコピーはかなり面倒です．

そこで，FM-7/11用の便利なディスク間ファイル転送プログラムを作りました．

写真1 全ファイルをコピー中！

```
R.

Filecopy Drive 0-->1 / Return, *, Filena
me =*

RT       /Copied
K        /Copied
RELOC    /Copied
KMON     /Copied
ST.OJ    /Copied
SPK8     /Copied
L        /Copied
SPB24    /Copied
KM.OJ    /Copied
SPB25    /Copied
OLDSTUP  /Copied
STARTUP  /Copied
SPFMAKE  /Copied
SPF      /Copied
```

file ◉特徴

●ファイル名を入力するだけで，マシン語プログラムでも，データ・ファイルでも，BASICプログラムでも（たとえプロテクトしてあっても），無条件にコピーできる．

● FMのDISK　BASICとコンパチなディレクトリとFATをつくるだけでなく，ディスクの書き込みフォーマットもまったくコンパチブルである．

●ディスケット１枚分の全ファイルをコピーするコマンドを持ち，ブランク・ディスケットにコピーした場合には，オリジナル・ディスケット上でとびとびのクラスタに存在していたファイルでも，ディレクトリの順に連続したクラスタに書き写される．これでFATやディレクトリの誤消去があった場合でもディスクの回復がしやすくなる．

●この機能を持ちながら，１クラスタにおさまっている．

file ◉プログラムの使い方

プログラムは正確に入れてください．このプログラムの入力ミスはディスクにとって致命的です．RUNすると〝Filecopy Drive 0−→1/Return, *, Filename=〟と聞いてくるので，このうち，１つが選べます．この段階ではディスケットの差し替えは自由です．

❶オリジナル・ディスケットをドライブ0（左側）に，コピーしたいディスケットをドライブ1に入れてください．ドライブ0にあるファイルの名前を入力した場合には，その後，次々と〝Filename=〟と聞いてくるので，1〜152個までの任意の個数のファイル名を入力して，最後に **RETURN** だけを入れてください（ファイル名は〝〟で囲まない）．

入力されたファイルがドライブ0のディスケットに実際にあるかどうか（60〜80行），それらの間に重複がないか（90〜100行），ドライブ1のディスケットのファイル名と一致していないか（110〜120行）をチェックし，ディスケットの空き領域を探し（130〜140行），FATエラーの有無を

表1　エラーメッセージ

Filename Too Long：入力したファイル名が８文字を超えている（このファイル名は無視される）．

(Filename)/Not Found：入力したファイル名がドライブ0のディスケットに見当たらない（このファイル名は無視される）．

(Filename)/Duplicated：入力したファイル名の間に同一のものが重複している（１つだけコピーされる）．

(Filename)/Already Exists：入力したファイル名と同一名のファイルがドライブ１のディスケットにある（このファイルはコピーされない）．

Disk Full：ドライブ１のディスケットの空き領域不足のため現在処理中，およびそれ以降のファイルはコピーできない（処理済のファイルは正常にコピーされる）．

(Filename)/FAT Error：ドライブ０のオリジナル・ファイルに関するFATにエラーがある（正常終了していない）ため，コピーを行なわない（他のファイルはコピーされる）．

Directory Full：ドライブ１のディスケットに空き領域はあるが，登録ディクトリイが232個以上であるため，コピーできない（このエラーは正常にDISK BASICを使っていれば決して出ないが，ディクトリ領域が何らかの理由で複製されたりした場合に出る．このエラー以前のファイルは正常にコピーされている）．

No File：ドライブ０のディスケットのディクトリが空であるため，コピーできない．

チェック(150行)したのち，クラスタ単位でファイルの本体をコピーします(160〜170行)．この後ディレクトリを書き換え(180行)，最後にまとめてドライブ1のFATを書き込んだ後，初期状態へ戻ります．

❷最初の〝Filename=〟に対して〝*〟を入力すると，ドライブ0のファイルを調べ，ドライブ1と重複がなければそれらを全部コピーします．2回目以降の〝Filename=〟に対して〝*〟を入力しても，このモードにはなりません．

❸最後に**RETURN**だけを入力するとプログラムは終わります．

◉終わりに

ささやかなBASICプログラムですが，使ってみると結構便利です．なお，プログラムに空白や**REM**文を1個でも入れると1クラスタにはおさまらなくなります．

◘参考文献

〝F-BASIC文法書〟富士通

ファイルコピープログラム・リスト

```
10 CLEAR9000:DEFSTRA-G:DEFINTH-R:DIMB(151),C(231),G(15),M(151):B=CHR$(0):C=CHR$(
255)
20 FIELD#0,128ASA(0),128ASA(1):DEFFNP(I)=(I+7)¥4:DEFFNQ(I)=((I+7)MOD4)*8+1
30 H=0:PRINT:PRINT"Filecopy Drive 0-->1 / Return, *, ";
40 LINEINPUT"Filename =",B(R):IFLEN(B(R))>8 THENBEEP:PRINT"Filename Too Long":GO
TO40 ELSEIFB(0)="" THENEND ELSEIFB(0)="*" THEN50 ELSE60
50 GOSUB200:R=L:IFL=0 THEN240 ELSEFORI=0TOL-1:B(I)=C(I):GOTO80
60 IFB(R)<>""ANDR<152 THENB(R)=LEFT$(B(R)+STRING$(8," "),8):R=R+1:GOTO40
70 GOSUB200:IFL=0 THEN240 ELSEFORI=0TOR-1:FORJ=0TOL-1:IFINSTR(C(J),B(I))>0 THENB
(I)=C(J) ELSENEXTJ:BEEP:PRINTB(I)+"/Not Found
80 NEXTI:IFR=1 THEN110
90 FORI=0TOR-2:IFLEN(B(I))=8 THENNEXTI:GOTO110
100 FORJ=I+1TOR-1:IFB(I)=B(J) THENB(I)=LEFT$(B(I),8):BEEP:PRINTB(I)+"/Duplicated
":NEXTI ELSENEXTJ,I
110 H=1:GOSUB200:IFL=0 THEN130
120 FORI=0TOR-1:FORJ=0TOL-1:IFINSTR(C(J),LEFT$(B(I),8))>0 THENB(I)=LEFT$(B(I),8)
:BEEP:PRINTB(I)+"/Already Exists":NEXTI ELSENEXTJ,I
130 PRINT:FORI=0TOR-1:H=LEN(B(I))-16:IFH<1 THENNEXTI:GOTO190 ELSEM=0
140 FORJ=0TOH-1:M(J)=INSTR(M+1,E,C):M=M(J):IFM=0 THENBEEP:PRINT:PRINT"Disk Full"
:GOTO190 ELSENEXTJ:MID$(B(I),15,1)=CHR$(M(0)-1)
150 L=ASC(RIGHT$(B(I),1)):IFL>199ANDL<>253 THENBEEP:PRINTLEFT$(B(I),1)+"/FAT Err
or":NEXTI:GOTO190
160 FORJ=0TOH-1:P=ASC(MID$(B(I),16+J)):IFJ<H-1 THENQ=7:A=CHR$(M(J+1)-1) ELSEIFL=
253 THENA=CHR$(L):GOTO180 ELSEQ=L-192:A=CHR$(L)
170 FORK=0TOQ:D=DSKI$(0,FNP(P),FNQ(P)+K):G(2*K)=A(0):G(2*K+1)=A(1):NEXTK:FORK=0T
OQ:LSETA(0)=G(2*K):LSETA(1)=G(2*K+1):DSKO$1,FNP(M(J)),FNQ(M(J))+K:NEXTK
180 MID$(E,M(J),1)=A:NEXTJ:M=INSTR(F,C):IFM=0 THENBEEP:PRINT:PRINT"Directory Ful
l" ELSED=DSKI$(1,1,(M-1)¥8+4):MID$(A(((M-1)MOD8)¥4),32*((M-1)MOD4)+1,32)=LEFT$(B
(I),15)+STRING$(17,B):DSKO$1,1,(M-1)¥8+4:MID$(F,M,1)=B:PRINTLEFT$(B(I),8)+"/Copi
ed":NEXTI
190 G(0)=B+STRING$(4,C)+LEFT$(E,123):G(1)=MID$(E,124)+STRING$(99,C):LSETA(0)=G(0
):LSETA(1)=G(1):DSKO$1,1,1:R=0:GOTO30
200 D=DSKI$(H,1,1):E=MID$(A(0),6)+LEFT$(A(1),29):L=0:F=STRING$(232,C)
210 FORI=4TO32:D=DSKI$(H,1,I):FORJ=0TO1:FORK=0TO3:D=MID$(A(J),32*K+1,15):IFASC(D
)=0 THEN230 ELSEIFASC(D)=255 THENRETURN ELSEN=ASC(MID$(D,15)):MID$(F,(I-4)*8+J*4
+K+1,1)=B
220 N=N+1:D=D+CHR$(N):N=ASC(MID$(E,N)):IFN>151 THEND=D+CHR$(N):C(L)=D:L=L+1 ELSE
220
230 NEXTK,J,I:RETURN
240 BEEP:PRINT"No File":END
```

RANDOM BOX

『TotoiseをKで』の演算精度をあげる

■横井幸彦

Kコンパイラのtortoiseの機能制限である，精度を実用上問題のないようにできます．右のように変更，追加します．

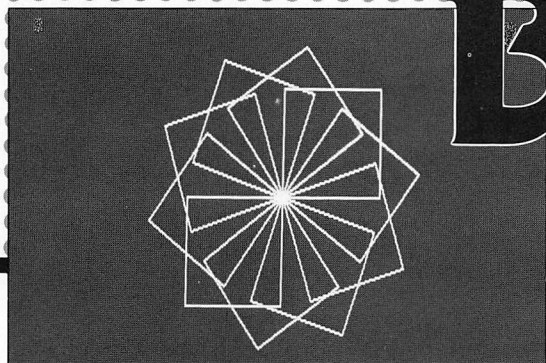
「内容」三角函数の精度はそのままで，乗除算の桁落ちをふせぎ，疑似的に小数第2位まで有効にしただけです。しかし，これだけで360(FR)は円になります．また縦横の比率は1/．4495により近い56/25に変更してあります．

変更・追加リスト

```
1300 'ACC=0;DIR=0;CLR=7;X=320;Y=100;XX=0;YY=0;W8=0
2460 'pushs;push[ACC];push[DIR];push[CLR]push[X];push[Y]push[XX];push[YY];goto e
vcdr
2470 'pulls;YY=pull[];XX=pull[];Y=pull[];X=pull[];CLR=pull[];DIR[];ACC=pull[];go
to evcdr
2540 'fome;X=320;Y=100;XX=0;YY=0;;GOTO evcdr
2780 'newXY;if W8=0 THEN
2785 '   SI=SIN[DIR];DBL1=0;DBS[ABS(SI)];DBM[STEP];DBM[56];DBD[25];XT=ANS[]*SGN(S
I)
2790 '    XF=UP[XT];XG=DOWN[XT];X=X+XF;XX=XX+XG;X=X+UP[XX];XX=DOWN[XX]
2800 '    YT=cos[DIR]*STEP;YF=UP[YT];YG=DOWN[YT];Y=Y-YF;YY=YY-YG;Y=Y+UP[YY];YY=DO
WM[YY];RETURN;FI
2951 'UP;DM=%1/100;RETURN
2952 'DOWN;DM=%1-%1/100*100;DM=DM;RETURN
2953 'DBS;DBL2=%1;DBL1=DBL1+DBL2/100;DBL2=DOWN[DBL2];RETURN
2954 'DBM;MD=%1;DBL1=DBL1*MD;DBL2=DBL2*MD;DBS[DBL2];RETURN
2955 'DBD;DD=%1;DO=DBL1/DD;RD=DBL1-DO*DD;DBL2=(DBL2+RD*100)/DD;DBL1=DO;RETURN
2956 'ANS;DD=DBL1*100+DBL2;RETURN
```

FM 7/11 LOGOのサブセットTortoiseをFMに移植

TortoiseをBASICで

■H・KOGUCHI

Tortoiseは，人工知能研究用に開発された，“Lisp”の派生語であり，最近評判の高い，“Logo”のグラフィック機能を取り出した超サブセット言語です．ここでは，I/O83年1，2月号のS.tanaquax氏のPascal版を，BASICで移植してみました．BASICインタープリンタで記述したため，甚だ実行速度が遅いのですが，フルセットを購入する費用を考えれば，再帰呼びだしも可能な，本Tortoiseは充分reasonableなものだと思います．

使用法その他

基本

コマンド表（**表1**）を見てください．基本的な使い方は，これでわかると思いますが，実際の実行の様子を6角形の作図を例として，**図1～4**に示します．これで，だいたいのプログラム手順がのみこめるでしょう．

再起

さて，もうひとつ，Tortoiseの特長である再帰的図形を描く例として，2次の凹クロス・ステッチを描くプログラムと，実行例を図7に示します．再帰について書くには，余りに紙面が足りないので，最後に挙げる参考文献や，Pascalの入門書などを読んでみてください．

再帰呼び出しは，その構造からしてなかなかに面白く，アイデア次第でいろいろと楽しい図形が描けます．有名なものには，ドラゴン曲線，ヒルベルト曲線，シェルピンスキー曲線などがあります．

エラー関係

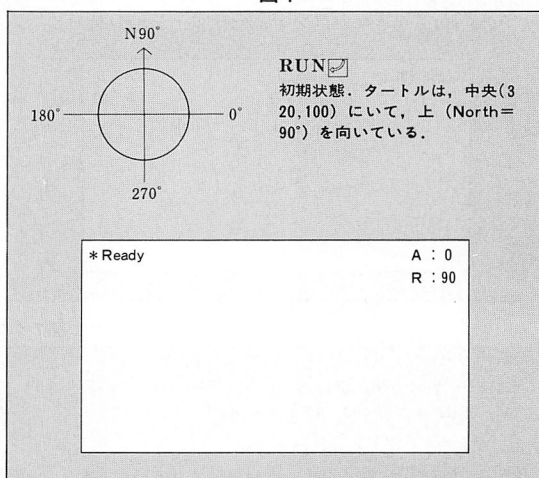
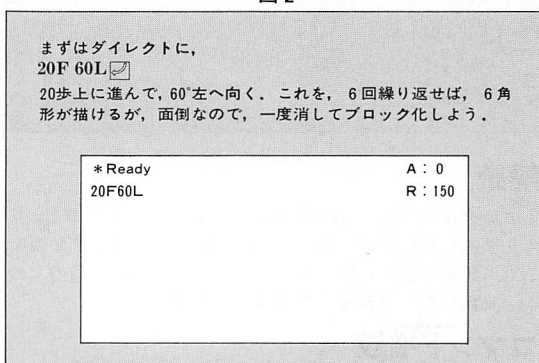
表2にエラーメッセージと対処法を示しました．特に手続き名は，指定範囲内（キャラクタコード&H5B～&H70）の中で使ってください．原則として，英小文字だけで充分間に合うはずです．BASIC側でエラーが出てしまった場合は，残念ですが最初からやり直しです．

エディタ

図1

図2

まずはダイレクトに，
20F 60L
20歩上に進んで，60°左へ向く．これを，6回繰り返せば，6角形が描けるが，面倒なので，一度消してブロック化しよう．

```
＊Ready                A：0
20F60L                 R：150
```

エディットに関しては，BASICのエディタ機能がそのまま使えるので，大変便利になっています．手続きの内容を変更したいときなど，新しく書き換えずに“／”コマンドでリストを取って，頭の部分をDnとし，以下重ね書きや削除，挿入などすれば良いのです．

プログラムについて

図3 初期化

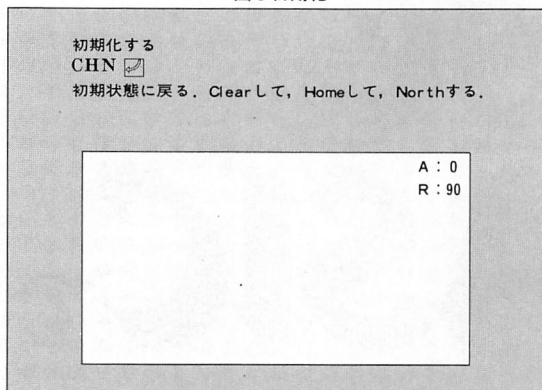



図5 手続化

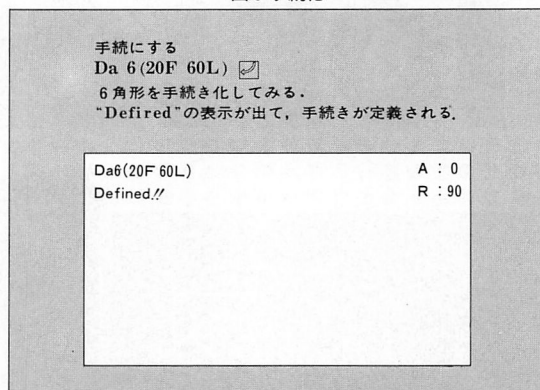



図4 ブロック化

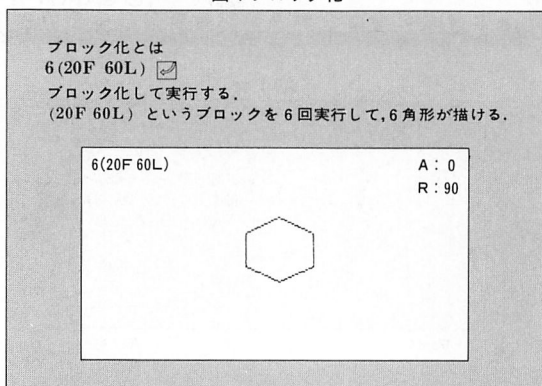
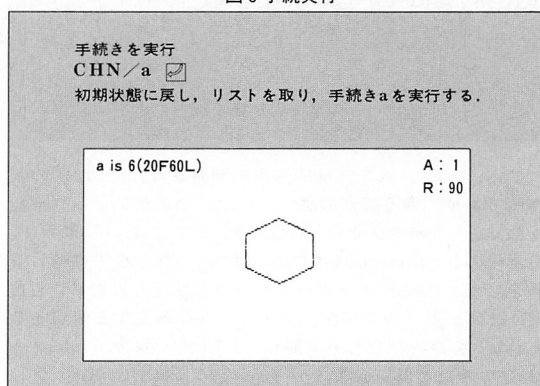
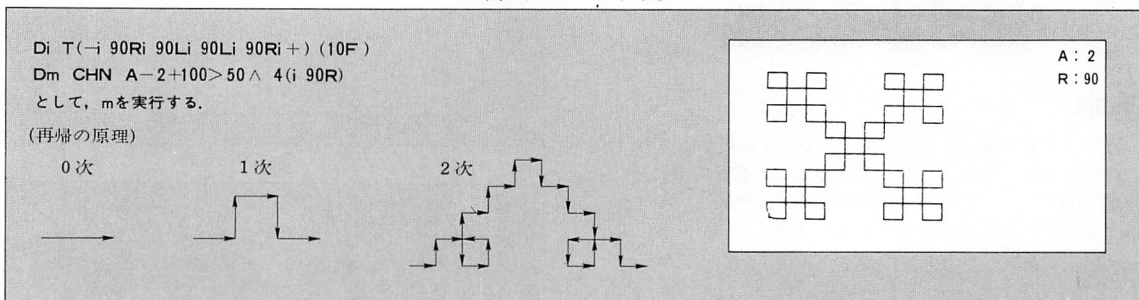



図6 手続実行

図7 クロス・ステッチ

解読

プログラムはPascal風にモジュール化してあるので，リストを追って行けば，特にフローチャートを起こす必要もなく，解読できると思います．メンテナンスの手助のために，変数リスト（**表3**）を付けておきます．

コマンド増設

PAINTなどのコマンドを増設したい方は，Procedure Evalの部分にIFコマンドTHEN処理モジュールのように定義すれば，好きなように増設することができます．ただし，引数類の変化に充分気をつけてください（特に再帰的呼び出しする場合）．

再帰呼び出し

Procedure Eval, Test, Sense, Userにおいて，再帰的呼び出しをしている部分があります．この部分はBASICでうまく書くために工夫がしてあります．EC, P, FPなどをスタックとして用意した配列に，PUSHして，再帰呼び出した後リターンしてきた時点で，PULL，スタック・ポインタをデクリメントする，という方法をとっています．マシン語で書くのと同じようなものです．

実行速度

実行速度の遅さを，いくらかでも解消したい方は，まず一切のREM文を消去し，リストの空白部分をみんなつめてしまってみてください（この際，GOSUBなどの飛び先に注意）．いくらかは，速くなるでしょう．

また，実行速度を遅くしている他の原因としては，実行中の行の直前に飛ばすような**GOTO**文（ほとんどない）や，文字列処理中のガベージコレクション（ここでは，特にCM＄の処理），プログラムの流れが直線的でないこと，EVALにおける判断の時間，再帰呼び出しを行なっていることなどがあると思いますが，筆者は所詮BASICで記述したもの

表1　コマンド

コマンド		意味	説明	使用例・使用法
動作	F	Forward	一歩前進（足跡を残して）.	nF, nは画面内におさまる範囲に限る.
	J	Jump	一歩ジャンプ（足跡残さず）.	nJ, nは画面内におさまる範囲に限る.
	C	Clear	画面を消去（足跡消す）.	C, A, Rotは変化しない.
位置・方向	R	Right	右方向に1度回転.	90R, 90°右に回転.
	L	Left	左方向に1度回転.	80L, 80°左に回転.
	^		タートルの垂直位置を設定.	20^で, 下から20歩の位置に設定.
	>		タートルの水平位置の設定.	40> で, 左から40歩の位置に設定.
	H	Home	タートルを中心に置く.	H, (320, 100)の点にタートル位置を設定.
	N	North	タートルを上方向（90°）を向かせて置く.	N, 90°方向にタートルが向く.
変数・演算	A	Accum later	グローバル変数（A≧0）.	AF, AJなどで, それぞれAの値だけ動作する.
	+	increment	Aに1加える.	2+, Aの値を2インクリメントする.
	−	decrement	Aから1減らす.	3−, Aの値を3デクリメントする.
条件判断	T	Test	T(a)(b). A> 0のときaを, そうでないときbを実行する.	T（90R）（60F）
	S	Sense	S(a)(b). 進行方向に既に足跡があれば, aを, そうでないときbを実行.	S（10F　90R）（10F　90L）
	?	Random	?(a)(b). 乱数によってaかbを実行する.	?（50J　60L）（10F　20R）
定義	D	Define	手続きを定義する. ※ただし, 手続き名は, 英小文字および, いくつかの特殊記号. キャラクタ・コード　&H5B～&H70	Da 4（10F　90L） 一辺が, 10歩の左巻き正方形を描く手続きが定義される.
繰返りし	( )		n(a). aというブロックをn回繰り返す.	6（20F　60L）で, 20F60Lというブロックを6回繰り返す.
その他のユーティリティー	／	List	定義された手続きを, 画面に表示する.	／（スラッシュ）で, すべての手続きを表示する.
	%	Llist	定義された手続きを, プリンタに印字する.	%（パーセント）で, すべての手続きを印字する.
	＊	Copy	描かれた図形（画面全体）をハードコピーする.	＊（アスタリスク）でHARDC2が実行される。
	#	Save	定義された全手続き（プログラム）を, カセットにセーブする.	#（シャープ）で,“FILE NAME?”に続き, ファイルネームを打ち込んで ⏎ する.
	$	Load	Saveコマンドで, セーブされた手続き（プログラム）を, カセットからロードする. ※ファイル名を照合し見つかるまで,“Skipf:ファイル名”を表示してスキップします.	$（ダラー）で,“File Name?”に続き, ファイル・ネームを打ち込んで ⏎ する.
	CTRL + G	Break	実行を中断します. ※ただし, タイミングによって, BASICの側でエラーが発生します.	ブザーが鳴り,“Abort”の表示が出るまで CTRL + G を押し続ける.
	@	End	Tortoiseのシステムを終了して, BASICに戻ります.	@（アットマーク）で, END文を実行します.
注	␣	Space	プログラム中のスペース	プログラム中のスペースは無視されます.

表2　エラーメッセージ

エラーメッセージ	内容	対処法
*Check Parenthesis	カッコの対応数が合っていない.	カッコの数を合わせて, やり直してください.
*Illegal Function Name	手続き名がない.	手続き名を, 正しくつけて, やり直してください.
その他のメッセージが出た場合	BASIC側のエラーです. 例）LINEの画面範囲からのオーバー.	もう1度RUNして, 最初からやり直してください.

表3　変数表

変数名	内容	
INITX, INITY	トータスのいる最初の位置(320, 100).	
ACC	グローバル変数Aの内容.	
ROT	タートルの向いている方向.	
X, Y	タートルの現在位置.	
ABORT	実行中断のためのアボート・フラグ. CTRL+Gが押されたとき1, 通常は0.	
NF	定義された手続きの数.	
S$(0)	ユーザーが入力した文字列がそのまま入る. 入力直後に“.”が後尾につけられる.	
S$(1～n)	ユーザーが定義した手続きが入っている。	
FP	現在実行中の手続きを示すポインタ.	すべてEVALのローカル変数
P	現在実行中のコマンドを示すポインタ.	
EC	コマンドを何回実行するかを記憶している.	
CM$	現在実行中のコマンド格納している.	
S	再帰的呼出しの際のスタック・ポインタ.	
その他の変数	手続き内でのローカル変数のため解読は容易.	

だから, と手を付けていません. 興味がある方は, いろいろやってみてください.

欠点

❶クリッピング処理をしていないため, タートルが（0, 0）-(639, 199)の範囲を出るとBASIC側でエラーが出てしまいます.

❷ CTRL + G で実行を中断する際, タイミングによってはBASIC側でエラーが発生してしまいます.

❸定義された手続きは, 変数をクリアする以外消去する方法がありません.

BASICでシステムを記述するなどというのは, 邪道もいいところかも知れません. しかし, BASICでも, できるということにはなります. 実行速度の遅さは大目に見てやって, このTortoiseもどうか可愛がってください.

なお, FM-11で実行するときはSCREEN0でRUNしてください.

参考文献

1) S.tanaquax; “Tortoise(1), (2)”, I/O '83年1,2月号

2) 石田則道; “ロボット言語で怪物曲線を描こう”, bit, vol. 65, No. 10

Tortoise BASICプログラム・リスト

```
10 ' TORTOISE FOR FM-
20 ' TRANSFERRER : H.KOGUCHI
30 '        DATE : From  APRIL 29 / 1983To JUN 18 / 1983
40 '
50 '
60 '
70 '
80 '
90 '
100 'PROGRAM TORTOISE
110 'CONST
120      INITX=320:'          (* CENTER OF SCREEN *)
130      INITY=100
140      MAXFUNC=25:'         (* MAXMUM USER-DEFINABLE FUNCTIONS *)
150      PI#=1.74533E-02:'    (* PI/180 *)
160 'VAR
170 '     ACC   : INTEGER ; (* CONTENTS OF ACC *)
180 '     S     : ARRAY [0..MAXFUNC] OF STRING
190 '     NF    : INTEGER;  (* NUMBER OF DEFINED FUNCTION *)
200 '     X     : REAL ;    (* X COORD. *)
210 '     Y     : REAL ;    (* Y COORD. *)
220 '     ROT   : INTEGER ; (* ROTATION FACTOR *)
230 '     ABORT: BOOLEAN ; (* ABORT FLAG *)
240 DIM S$(25),EC(500),FP(50),P(50),LP(50):LOOP=1:GOTO21000
300 'PROCEDURE INITROBOT (* INITIALIZER OF ROBOT LANGAGE *)
305      WIDTH80
310      ACC=1
320      NF=0
330      X=INITX
340      Y=INITY:LINE(0,0)-(INITX,INITY),OR,0
350      ROT=90
360      PRINT CHR$(12)
370      PRINT "*Ready"
380  RETURN
400 'PROCEDURE DISPACC (* DISPLAY THE CONTENTS OF ACC ON CRT *)
405      V=CSRLIN
410      LOCATE70,0:PRINT"A: ";
420      SACC$=STR$(ACC)
425      SACC$=SACC$+"  "
430      PRINT SACC$:PRINT
435      LOCATE0,V
440  RETURN
500 'PROCEDURE DISPROT (* DISPLAY THE CONTENTS OF ROT ON CRT *)
505      V=CSRLIN
510      LOCATE70,1:PRINT"R: ";
520      SACC$=STR$(ROT)
525      SACC$=SACC$+"  "
530      PRINT SACC$:PRINT
535      LOCATE0,V
540  RETURN
600 'PROCEDURE GETLINE (* READ ONE LINE OF COMMAND FROM CONSOLE *)
610      LINEINPUT T$:P=0:EC=1
620      S$(0)=T$+".":'(* ADD TERMINATER *)
630      (* CHECK NESTING LEVEL *)
640       NEST=0:I=1:FUN$=""
650       WHILE FUN$<>"."
660            FUN$=MID$(S$(0),I,1)
670            I=I+1
680            IF FUN$="(" THEN NEST=NEST+1 ELSE IF FUN$=")" THEN NEST=NEST-1
690       WEND
695       IF NEST<>0 THEN PRINT"*Check Parenthesis !"+CHR$(7):S$(0)="...."
697 RETURN
700 'FUNCTION READCOM (* READ ONE COMMAND*)
710      P=P+1
720      REDCOM$=MID$(S$(FP),P,1)
730 RETURN
800 'PROCEDURE  MINUS (* DECREMENT ACC BY ONE *)
810      ACC=ACC-EC
820      IF ACC<0 THEN ACC=0
830      GOSUB400
840      EC=1
850 RETURN
900 'PROCEDURE PLUS (* INCREMENT ACC BY ONE *)
910      ACC=ACC+EC
920      GOSUB400
930      EC=1
940 RETURN
1000 'PROCEDURE NUMBER (* GET A NUMBER INTO EC *)
1010      NC=0
1020      CH$=MID$(S$(FP),P,1)
1030       WHILE (ASC(CH$)>=48)AND(ASC(CH$)<=57)
1040         NC=NC*10+ASC(CH$)-ASC("0")
```

```
1050          P=P+1
1060          CH$=MID$(S$(FP),P,1)
1070         WEND
1080       EC=NC
1090       P=P-1
1095 RETURN
2000 'PROCEDURE FWARD (* MOVE TURTLE LEAVING A LINE *)
2005       LINE-(CINT(X),CINT(Y)),OR,0
2010       X=COS(ROT*PI#)*EC*2+X
2020       Y=Y-SIN(ROT*PI#)*EC
2030       COLOR7:LINE-(CINT(X),CINT(Y)),PSET
2040       EC=1
2050 RETURN
3000 'PROCEDURE JUMP (* MOVE TURTLE WITHOUT LEAVING A LINE *)
3010       X=COS(ROT*PI#)*EC*2+X
3020       Y=Y-SIN(ROT*PI#)*EC
3030       EC=1
3040 RETURN
4000 'PROCEDURE LEFT (* TURN LEFT *)
4010       ROT=(ROT+EC) MOD 360
4020       GOSUB500
4030       EC=1
4040 RETURN
5000 'PROCEDURE RIGHT (* TURN RIGHT *)
5010       ROT=(ROT-EC+360) MOD 360
5020       GOSUB500
5030       EC=1
5040 RETURN
6000 'PROCEDURE SKIPPER (* SKIP PARENTHESIS *)
6010       WHILE CM$<>"("
6020           GOSUB700:CM$=REDCOM$
6030       WEND
6040       NEST=1
6050       WHILE NEST>0
6060           GOSUB700:CM$=REDCOM$
6070           IF CM$="(" THEN NEST=NEST+1ELSE IF CM$=")" THEN NEST=NEST-1
6080       WEND
6090 RETURN
7000 'PROCEDURE TEST (* IF ACC IS NON-ZERO, EXECUTENEXT PARENTHESIS
7010                      OTHERWISE, EXECUTE NEXTNEXT PARENTHESIS *)
7020       WHILE EC>0
7030           IF ACC<>0 THEN 7040 ELSE 7060
7040               LP=P+1:S=S+1:EC(S)=EC:FP(S)=FP:P(S)=P:LP(S)=LP:EC=1:P=LP
7050               GOSUB20000:EC=EC(S):FP=FP(S):P=P(S):LP=LP(S):S=S-1:GOTO7100
7060               S=S+1:P(S)=P:EC(S)=EC:FP(S)=FP:P(S)=P
7070               GOSUB6000
7080               LP=P+1:EC=1:P=LP
7090               GOSUB20000:EC=EC(S):FP=FP(S):P=P(S):S=S-1
7100           EC=EC-1
7110       WEND
7120        GOSUB6000
7130        GOSUB6000
7140        EC=1
7150  RETURN
8000 'PROCEDURE DEFUN (* DEFINE FUNCTIONS *)
8010       T$=S$(0)
8020       IF MID$(S$(0),2,1)=" " THEN8030 ELSE 8060
8030           PRINT"*Illegal Function Name !!"+CHR$(7)
8040           S$(0)="....":EC=0
8050           RETURN
8060        (* GET THE FUNCTION NAME *)
8070        GOSUB700:FUN$=REDCOM$
8080        FOUND=0
8090        FOR I=1 TO NF
8100           IF FUN$=LEFT$(S$(I),1) THEN 8110 ELSE 8130
8110               S$(I)=MID$(T$,2,LEN(T$)-1)
8120               FOUND=1
8130        NEXT I
8140        IF FOUND=0 THEN 8150 ELSE 8170
8150           NF=NF+1
8160           S$(NF)=MID$(T$,2,LEN(T$)-1)
8170        PRINT"*Defined !!"+SPACE$(15)
8180        S$(0)="....":EC=1
8190 RETURN
9000 'PROCEDURE SENSE (* IF NEXT DOT IS PAINTED, EXECUTE NEXT PARENTHESIS
9010                      OTHERWISE, EXECUTE NEXT NEXT PARENTHESIS *)
9020       WHILE EC>0
9030           X=COS(ROT*PI)+X
9040           Y=Y-SIN(ROT*PI)
9050           IF POINT(CINT(X),CINT(Y)) THEN 9060 ELSE 9080
9060                 LP=P+1:S=S+1:EC(S)=EC:FP(S)=FP:P(S)=P:P=LP:EC=1
9070                 GOSUB20000:EC=EC(S):FP=FP(S):P=P(S):S=S-1:GOTO9115
9080                 S=S+1:P(S)=P:EC(S)=EC:FP(S)=FP:P(S)=P
```

Tortoise BASICプログラム・リスト

```
9090                GOSUB6000
9100                LP=P+1:EC=1:P=LP
9110                GOSUB20000:EC=EC(S):FP=FP(S):P=P(S):S=S-1
9115        EC=EC-1
9120      WEND
9130      GOSUB6000
9140      GOSUB6000
9150      EC=1
9160 RETURN
10000 'PROCEDURE RND (* EXECUTE TWO PARENTHESIS AT RANDOM *)
10010      WHILE EC>0
10020          IF RND(8)<.5 THEN 10030 ELSE 10050
10030               LP=P+1:S=S+1:EC(S)=EC:FP(S)=FP:P(S)=P:P=LP:EC=1
10040               GOSUB20000:EC=EC(S):FP=FP(S):P=P(S):S=S-1:GOTO10090
10050              S=S+1:P(S)=P:EC(S)=EC:FP(S)=FP:P(S)=P
10060               GOSUB6000
10070               LP=LP+1:EC=1:P=LP
10080               GOSUB20000:EC=EC(S):FP=FP(S):P=P(S):S=S-1
10090          EC=EC-1
10100      WEND
10110      GOSUB6000
10120      GOSUB6000
10130      EC=1
10140 RETURN
11000 'PROCEDURE HOME (* PLACE TURTLE AT HOME POSITION *)
11010      IF EC>0 THEN X=INITX:Y=INITY
11020      EC=1
11030 RETURN
12000 'PROCEDURE NORTH (* SET DIRECTION TO UPWARD *)
12010      IF EC>0 THEN ROT=90:GOSUB500
12020      EC=1
12030 RETURN
13000 'PROCEDURE CLS (* CLEAR CRT SCREEN *)
13010      IF EC>0 THEN CLS:GOSUB400:GOSUB500
13020      EC=1
13030 RETURN
14000 'PROCEDURE LIST (* LIST ALL FUNCTIONS TO CRT *)
14010      IF S$(1)<>"" THEN GOTO14020 ELSE 14060
14020           FOR I=1 TO NF
14030              PRINT LEFT$(S$(I),1)+" IS "+MID$(S$(I),2,LEN(S$(I))-2)
14040           NEXT
14050      EC=1
14060 RETURN
15000 'PROCEDURE LLIST (* LIST ALL FUNCTION TO PRINTER *)
15010      OPEN "O",#1,"LPT0:"
15015      IF S$(1)<>"" THEN 15020 ELSE 15050
15020        FOR I=1 TO NF
15030          PRINT# 1,LEFT$(S$(I),1)+" IS "+MID$(S$(I),2,LEN(S$(I))-2)
15040        NEXT I
15050      EC=1:CLOSE1
15060 RETURN
15100 'PROCEDURE SAVE (* SAVE ALL FUNCTION TO CASSTTE *)
15108      INPUT"FILE NAME";FI$:FI$=FI$+","
15110      OPEN"O",#2,"CAS0:"
15112      PRINT#2,FI$
15120      IF S$(1)<>"" THEN 15125 ELSE 15160
15125      NF$=HEX$(NF)+",":PRINT#2,NF$
15130        FOR I=1 TO NF
15140          PRINT#2,S$(I)
15150        NEXT I
15160      EC=1:CLOSE#2
15170 RETURN
15200 'PROCEDURE LOAD (* LOAD ALL FUNCTION TO CASSTTE *)
15208      INPUT"File Name";FI$:FI$=FI$+","
15210      OPEN"I",#2,"CAS0:"
15212      INPUT#2,FC$
15213      IFFC$<>FI$THENPRINT"Skip: "+LEFT$(FC$,LEN(FC$)-1):GOSUB15280:GOTO15210

15214      PRINT"Found: "+LEFT$(FC$,LEN(FC$)-1)
15215      INPUT#2,NF$:NF=VAL(NF$)
15230        FOR I=1 TO NF
15240          LINEINPUT#2,S$(I)
15250        NEXTI
15260      EC=1:CLOSE#2
15270 RETURN
15280 INPUT#2,NF$:NF=VAL(NF$)
15281 FOR I=1 TO NF:LINEINPUT#2,DMY$:NEXTI
15282 CLOSE#2:RETURN
16000 'PROCEDURE COPY (* TAKE A HARDCOPY OF THE GRAPHIC SCREEN *)
16010      HARDC2
16020 RETURN
17000 'PROCEDURE USER (* EXECUTE USER DEFINED FUNCTIONS *)
17010      FOUND=0
```

```
17020      FOR I=1 TO NF
17030         IF CM$=LEFT$(S$(I),1) THEN FOUND=1:J=I
17040      NEXT
17050      IF FOUND THEN 17060 ELSE 17100
17060         WHILE EC>0
17070            S=S+1:EC=EC-1:EC(S)=EC:FP(S)=FP:FP=J:P(S)=P:P=1:EC=1
17080            GOSUB20000:EC=EC(S):FP=FP(S):P=P(S):S=S-1
17090         WEND
17100      EC=1
17110 RETURN
18000 'PROCEDURE SETY (* SET VERTICAL POSITION TO EC *)
18010      Y=199-EC
18020      EC=1
18030 RETURN
19000 'PROCEDURE SETX (* SET HORISONTAL POSITION TO EC *)
19010      X=EC
19020      EC=1
19030 RETURN
20000 'PROCEDURE EVAL (* MAIN EVALUATER FOR TORTOISE LANGAGE *)
20010      GOSUB400
20020      GOSUB500
20030      WHILELOOP=1
20035 'LOCATE1,0:PRINTP;FP
20040          IF ABORT THEN RETURN
20050 '          (* IF CTRL-G IS PRESSED THEN ABORT *)
20060          IF INKEY$=CHR$(7) THEN ABORT=1
20070 '        (* GET NEXT COMMAND *)
20080          GOSUB700:CM$=REDCOM$
20090 '          (* CASE COM$ OF *)
20100          IF CM$="A" THEN EC=ACC
20110          IF CM$="-" THEN GOSUB800
20120          IF CM$="+" THEN GOSUB900
20130          IF CM$="(" THEN20140 ELSE20220
20140                      WHILE EC>0
20150                           IF ABORT THEN EC=0
20160                           S=S+1:EC=EC-1:EC(S)=EC:FP(S)=FP:P(S)=P:EC=1
20170                           GOSUB20000:EC=EC(S):FP=FP(S):P=P(S):S=S-1
20180                      WEND
20190              P=P-1
20200              GOSUB6000
20210              EC=1:GOTO20440
20220          IF CM$=")" THEN RETURN
20230          IF CM$="F" THEN GOSUB2000
20240          IF CM$="J" THEN GOSUB3000
20250          IF CM$="R" THEN GOSUB5000
20260          IF CM$="L" THEN GOSUB4000
20270          IF CM$="T" THEN GOSUB7000
20280          IF CM$="D" THEN GOSUB8000
20290          IF CM$="S" THEN GOSUB9000
20300          IF CM$="?" THEN GOSUB10000
20310          IF CM$="H" THEN GOSUB11000
20320          IF CM$="N" THEN GOSUB12000
20330          IF CM$="C" THEN GOSUB13000
20340          IF CM$="." THEN RETURN
20350          IF CM$="/" THEN GOSUB14000
20360          IF CM$="%" THEN GOSUB15000
20370          IF CM$="*" THEN GOSUB16000
20380          IF CM$=" " THEN REM
20390          IF CM$="^" THEN GOSUB18000
20400          IF CM$=">" THEN GOSUB19000
20405          IF CM$="#" THEN GOSUB15100
20406          IF CM$="$" THEN GOSUB15200
20410          IF CM$="@" THEN END
20420          IF ASC(CM$)>=48 AND ASC(CM$)<=57 THEN GOSUB1000
20430          IF ASC(CM$)>=&H5B AND ASC(CM$)<=&H7D THEN GOSUB17000
20440      WEND
20450 RETURN
21000 '(***** MAIN ROUTINE *****)
21010      GOSUB300
21020      RANDOMIZE(15000)
21030      WHILE LOOP=1
21040          ABORT=0:'       (* NOTABORT *)
21045          GOSUB400:GOSUB500
21050          GOSUB600:'(* GET A COMMAND LINE *)
21060          GOSUB20000:'(* EVALUATE THE LINE *)
21070          IF ABORT THEN PRINT"*Abort"
21080      WEND
21090 END
```

TortoiseをKコンパイラで

■小松　仁

LOGOのタートル・グラフィックスの概念でプログラミングして，図形を描くことができるインタープリタです．Kコンパイラで作ったので，ドラゴン曲線やヒルベルト曲線などの図形を数秒～数10秒で描きます．

このインタープリタは，LISPインタープリタと同じ構造で設計しています．

この言語は，1977年にLichen Wangという人が〝ロボット言語〟として発表したものですが，I/O誌では〝Tortoise〟という名前でおなじみのものです．ここではTortoiseを縮めた〝TOTO〟という名前にします．

このコマンドは，すべて１文字であるため，プログラムが記号の羅列になって読みにくいという欠点がありますが，逆に文字数が少なくて扱いやすいので，マニア向けの言語だと思います．コマンドを〝進〟，〝右〟，〝左〟などの漢字に置き換えると，漢字調の日本語LOGOになりそうです．

システム構成

人間と会話するCCP (Console Command Processor) 部はF-BASICで，TOTO言語を解釈実行するインタープリタはKコンパイラで記述しています(図１)．

TOTOには，関数の再帰呼び出しとカッコを使う構造化の機能があるので，実行対象のプログラムはリスト構造に変換してから解釈実行しています．

Kコンパイラは予想以上に効率が良くて，解釈実行の部分をマシン語にしてテストしたところ，画面一杯のドラゴン曲線で約30％しか速くなりません(FM-8)．マシン語化するよりFM-7の方が勝ります．

CCPコマンド

TOTO のシステムを操作する CCP コマンドは表１のとおりです．CCPはキーボーボから入力されたデータを解析して，それがCCPコマンドであれば，指示された動作を行ない，CCPコマンドでなければ，TOTO 言語のプログラムであるとみなして，格納し実行します．

入力したデータの先頭が＇(クォート) の場合はコメント文として扱います．

プログラムの修正は，**LIST**コマンドで表示してから，BASICのスクリーン・エディット機能と同様にカーソルを移動して行ないます．ただし１回に１行の修正しかできません．

図１　TOTOシステムの構成

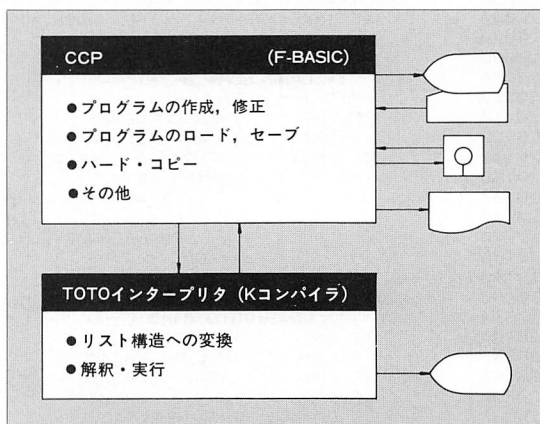



表１　CCPコマンド

形　式	意　　味
―	番号１から終わりまで実行する．
LIST	(F-BASICのLISTと同じ)
NEW	(F-BASICのNEWと同じ)
DEL␣*n*	番号*n*のプログラムを削除する．
INS␣*n*	番号*n*の位置にプログラムを挿入する．現在*n*番のプログラムは*n*＋1番になる．
LOAD␣ファイル名	指定されたファイルの内容をロードして表示する．
RUN␣ファイル名	指定されたファイルの内容をロードして実行する．
SAVE␣ファイル名	プログラムを格納する．番号１のデータがコメント文でない場合，ファイル名をコメント文にしたデータを自動生成する．
SUBMIT␣ファイル名	指定されたファイルのデータをキーボードから入力したデータであるとみなす．ファイルのデータはすべてコメント文でなければならない．
FDUMP␣ファイル名	プリンタにリストする．
FILES	(F-BASICのFILESと同じ)
TRON	トレース・モードにする．150個のコマンドをトレースするとトレース・オフとなる．
HARDC0 HARDC1 HARDC2	(F-BASICのHARDC0と同じ)

表2　TOTO言語の文法規則

10進数をn，コマンドをCで表わす．または，これらの組み合わせ，たとえば50F90Rを式とする．	
1.動　作;	nC……コマンドCをn回実行する． AC……コマンドCをAccの内容で指定される回数だけ実行する．
拡張1;	Cのところで(式)と書くと式をn回実行する．
拡張2;	Cのところで関数名fと書くとプログラムで定義した関数をn回実行する．
2.関数定義;	Df(式)……コマンドDによって任意の文字fを関数名とし，その動作を式で定義する．プログラムの中に文字fがあると式が実行される．
3.判　断;	T(式1)(式2)……(Acc)>0であれば式1を実行し，(Acc)＝0であれば式2を実行する．

表3　TOTO言語のコマンド

コマンド	意味
F (Forward)	1歩前進．nFならばn歩前進．
J (Jump)	1歩ジャンプ．軌跡を残さない．
R (Right)	右まわりに角度の1度向きを変える．
L (Left)	左まわりに角度の1度向きを変える．
C (Clear)	画面クリア．
H (Home)	画面の中央に位置づけする．
N (North)	北向き(上方向)にする．
A (Accumulator)	繰り返し回数を(Acc)で指定する．
＋(Increment)	(Acc)＋1
－(Decrement)	(Acc)－1
%(Percent)	n%ならば(Acc)×n÷100
T (Test)	表2の文法規則参照．
D (Define)	表2の文法規則参照．
<(Push)	現在位置，方向，Acc，色をスタックに格納する．
>(Pull)	現在位置，方向，Acc，色をスタックから戻す．
#(色指定)	n#でカラーコードnの指定となる．
*(8方向)	北向きを0度とし，タテ，ヨコ，ナナメ，すなわち45度単位の動作モードにする．スピードが早くなる．
CC (newsモード)	クリア・コマンドの次の文字がCの場合，newsモードになり，コマンドE，W，Sが有効になる．
	右向きにする．N90Rと同等．
E (East)	左向きにする．N90Lと同等．
W (West)	下向きにする．N180Rと同等．
S (South)	n$の指定で現在位置から，カラーコード$n$でペイントする．境界色は現在のカラーモード，n，青の3色．
$ (Paint)	

㊟ %，<，>，*のコマンドは筆者が考案したものです．

TOTO言語の仕様

この言語は，1文字のコマンドと，コマンドの実行回数を指定する10進数を組み合わせプログラミングします．

たとえば，コマンド**F**はForwardの略で，**50F**とすれば50歩前進し，コマンド**R**はRightの略で，**90R**とすると90度右に向きが変わるので，**4(50F90R)**のプログラムは大きさ50の正方形を描きます．システムには，アキュムレータ(Acc)が1個あり，**A**と書くとAccの内容がコマンドの実行回数を指定することになります．

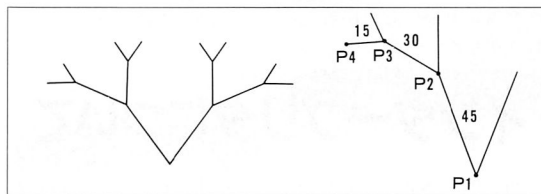
言語の文法規則を**表2**，コマンドを**表3**に示します．

TOTOプログラミング

このTOTO言語で描くことができる高級な図形には，

●ドラゴン曲線　●C曲線
●ヒルベルト曲線　●シェルピンスキー曲線

がありますが，これらの曲線は数学の学者が発明したものですので，プログラムは難解です．

図2　TREEの作図法

ここでは，やさしいTREE(**図2**)をとりあげ，さらにプログラムを改造して新しい図形を作る方法を説明します．

左端の枝から描くことにすると，P1を出発点にして，30度左に向いてAccの内容，ここでは45だけ進んでP2にいく(**30LAF**)．次にAccから15引いて，もう1度**30LAF**するとP3にきます．

これを繰り返すと，P4の次は(Acc)＝0となるので，終了にして，今度は1回まえのP3に戻って**30R**とすると，以前にP3にきたときと同じ向きになります．

```
Dt (T (30LAF15－tb30R) ( ))
Db (15＋180RAJ180L)
```

関数**b**は1回まえの位置に戻るサブルーチンです．(Acc)＝0のときは何もしないので，(　)の中は空です．この(　)は，右側に〝)″があるので，省略してもかまいません．

P1～P3の各点で枝分かれを作るには，向きを逆にして，

```
30LAF15－tb30L
```

を実行します．以上をまとめると，

```
Dt (T (30LAF15－tb30R30RAF15－b30L))
Db (15＋180RAJ180L)
```

となります．このプログラムでのAcc，方向，位置を1回まえに戻す操作はスタックを使用すると，

```
Dt (T (<39LAF15－t><30LAF15－t>))
```

となり，関数**b**は不要となります．

枝を描く関数ができたので，あとは出発点を定め，Accに初期値を入れて**t**を呼ぶと動きます．

```
D! (CHNA－180R90JN)
!45＋t
```

TREEのイメージ化

TOTOインタープリタは，絵を描くスピードが速いので，プログラムを修飾してユニークな図形を作るための道具として使うことができます．TREEのプログラムを改造，修飾するヒントを列挙します．

●色をつける．
●枝曲がりの角度，枝の長さを変更する．
●TREEの大きさ，あるいは位置を変えて複数個描く．
●枝を直線でなく，3角形や4角形などにする．
●枝の長さを%コマンドで変化させる．
●スタックを使わないで戻りの状態を不完全に，すなわち少しずらすようにする．

などなど，無限の可能性があります．かなり複雑な図形でも数10秒で描くことができます．

サンプル・プログラムの中の〝TREE.GC″は小学1年の

娘が色づけして，レンゲの花を表現した作品です．TREEプログラムで創作した図形だけをまとめた１冊の本が書店にありました．

インタープリタについて

TOTO言語では，変数が１つしか使えないので，LOGOのように関数へ引数を渡す機能が欲しくなりますが，これは言語仕様レベルの問題なので機能追加は大変です．

Push-pullコマンドのような単独コマンドは，容易に機能追加できます．

インタープリタの説明をするには，リスト，セル，アトム，CAR，CDRなどの基本概念から始めて，リスト構造への変換形式，EVALの定義とその動作など，LISP入門になりますので今回は省略します．

スピード優先のための手法を加えているので，ソース・リストは若干読みにくいかもしれません．

使用上の注意

再帰呼び出しでループ・ダウンすると，CPUのスタックを破壊して暴走するので危険です．この現象はたとえばTREEの**Dt**（**T**(……で**T**コマンドを入れ忘れると発生します．

一瞬のうちにダウンするので，ファイルはバック・アップをとっておき，デバッグ中はプログラム・セーブのたびにディスク・オフにすると安全です．

機能制限

1．座標位置および乗除算の演算は整数型，三角関数は10進２桁の精度のため，図形が変形することがあります．たとえば**360(FR)**は円にならず正方形になります．

2．実行させるTOTOプログラムが大きくなると，

```
String Too Long
```

エラーになります．

CCP実行中にF-BASIC上のエラーが発生したときは，

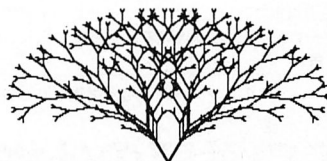
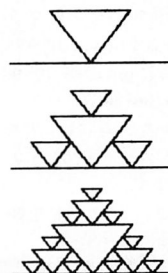
GOTO 2000 ⏎ または **PF4** ⏎

でCCPに戻ることができます．

参考文献

1)石田晴久：〝ドラゴン曲線やヒルベルト曲線を描くマイコン〟，bit，'78年６月号
2)石田則道：〝ロボット言語で怪物曲線を描こう〟，bit，'83年９月号
3)S.Tanaquax：〝tortoise〟，I/O，'83年１，２月号
4)渡辺誠太郎：〝Kコンパイラ用三角形関数サブルーチン〟，I/O，'83年10月号
5)コンピ：〝Kコンパイラ用グラフィック・サブルーチン〟，FM-7/8活用研究

実行例

```
01:'--- TREE.GC ---
02:Dt(T(<25Lf><25Rf>))
03:Df(cAF4-t)
04:Dc(4-T(4-T(4-T(4#)(7#)4+)(7#)4+)(3#)4+)
05:D!(CHNA-180R90JN)
06:!12+4(4+t)
```

```
01:'--- LEAF.1 ---
02:Dt(T(<90Lf>90Rf))
03:Df(AF<73%r>)
04:Dr(3-T(3+t))
05:D!(CHNA-180R95JN4#*)
06:!50+AFAFt
```

リスト１　TOTO CCP BASICプログラム・リスト

```
100   LOADM "TOTO.K2"
110   WIDTH 80,25 : CONSOLE 0,25,0,0 : CLEAR 1200,&H2FFF
120   '
130   PRINT : PRINT "   y - TOTO  for  FM    ( V-3.6 )" :'        TOTO.A2
```

リスト1　TOTO CCP BASICプログラム・リスト

```
140                                         (C) H.KOMATSU   84.01.04 no-1
150  PRINT
160  KEY 1,"TRON"+CHR$(13)         : KEY 6,"LOAD "
170  KEY 2,"LIST"+CHR$(13)         : KEY 7,"SAVE "
180  KEY 3,"-"+CHR$(13)            : KEY 8,"FILES"+CHR$(13)
190  KEY 4,"goto 2000"             : KEY 9,"**** "----unused
200  KEY 5,"RUN "                  : KEY10,"SUBMIT "
210  '
220  DEFINT A-Z : PSW=0 : TN=1 : FLNG$="():"+CHR$(34) : DV$="1:"
230  TNMAX=20 : DIM TX$(TNMAX+5)
240  '
250  BKFLAG=&H6C02 : KBFLAG=&H6C03 : TT=&H6C04
260  DEF FNTN$(I)=RIGHT$(STR$(I+100),2)+":"
270  '
280  POKE &H5000,&H34 : POKE &H5001,&H7F
290  POKE BKFLAG,&H00 : EXEC &H5000   ' call initializer ( K )
300  POKE TT,0 : GOTO 2010 '  no trace mode
310  '
1000 '#1000   call TOTO interpriter
1010 TXW$="("+TXW$+")" : TXA=VARPTR(TXW$)
1020 TXAH=INT(TXA/256) : TXAL=TXA-TXAH*256
1030     POKE &H4000,TXAH : POKE &H4001,TXAL
1040 IF PSW=0 THEN POKE BKFLAG,3 ELSE POKE BKFLAG,4 ' 4 --> reset FLP
1050     EXEC &H5000
1060 IF PEEK(TT)<>0 THEN PRINT : POKE TT,0 : RETURN ' trace off
1070 IF PEEK(KBFLAG)=ASC("C") THEN LOCATE 0,1
1080     RETURN
1090 '
2000 '#2000   CCP ( console command processor )
2010    IN$="" : WHILE IN$="" : PRINT FNTN$(TN); : LINEINPUT IN$ : WEND
2020    GOSUB 2040 : GOTO 2010
2030 'command analisys
2040    IF LEFT$(IN$,1)= "-"      THEN GOSUB 2390 : RETURN  ' KEY  3
2050    IF LEFT$(IN$,4)= "LIST"   THEN GOSUB 2490 : RETURN  ' KEY  2
2060    IF LEFT$(IN$,5)= "FDUMP"  THEN GOSUB 3020 : RETURN
2070    IF LEFT$(IN$,4)= "TRON"   THEN GOSUB 2460 : RETURN  ' KEY  1
2090    IF LEFT$(IN$,6)= "HARDC0" THEN HARDC      : RETURN
2100    IF LEFT$(IN$,6)= "HARDC1" THEN HARDC 1 :  : RETURN
2110    IF LEFT$(IN$,6)= "HARDC2" THEN HARDC 2 :  : RETURN
2120    IF LEFT$(IN$,3)= "NEW"    THEN TN=1 :     : RETURN
2130    IF LEFT$(IN$,4)= "SAVE"   THEN GOSUB 2750 : RETURN  ' KEY  7
2140    IF LEFT$(IN$,4)= "LOAD"   THEN GOSUB 2800 : RETURN  ' KEY  6
2150    IF LEFT$(IN$,3)= "RUN"    THEN GOSUB 2870 : RETURN  ' KEY  5
2160    IF LEFT$(IN$,4)= "SUBM"   THEN GOSUB 2920 : RETURN  ' KEY 10
2170    IF LEFT$(IN$,5)= "FILES"  THEN FILES DV$  : RETURN  ' KEY  8
2180    IF MID$(IN$,3,1)=":"      THEN GOSUB 2270 : RETURN  ' change
2190    IF LEFT$(IN$,3)= "INS"    THEN GOSUB 2290 : RETURN  ' insert
2200    IF LEFT$(IN$,3)= "DEL"    THEN GOSUB 2350 : RETURN  ' delete
2210 'store text
2220      TX$(TN)=IN$ : IF LEFT$(IN$,1)="'" THEN GOSUB 2260 : RETURN
2230 'exec TOTO
2240      TXW$=IN$ : GOSUB 1010 : GOSUB 2260 : RETURN
2250 'count TN
2260    TN=TN+1 : IF TN<TNMAX THEN RETURN ELSE PRINT "*** text full" : RETURN
2270 'change
2280    N=VAL(IN$) : TX$(N)=MID$(IN$,4) : GOSUB 2490 : RETURN ' list
2290 'insert
2300    N=VAL(MID$(IN$,4)) : IF N>0 THEN 2320
2310      INPUT "input line number---",IN$ : N=VAL(IN$) : IF N=0 THEN 2660
2320    PRINT FNTN$(N); : LINEINPUT IN$ : IF IN$="" THEN 2660
2330    FOR I=TN-1 TO N STEP -1 : TX$(I+1)=TX$(I) : NEXT : TX$(N)=IN$ :
        GOSUB 2260 : GOSUB 2490 : RETURN
2340 'delete
2350    N=VAL(MID$(IN$,4)) : IF N=0 THEN 2660
2360    FOR I=N TO TN-1 : TX$(I)=TX$(I+1) : NEXT
2370    TN=TN-1 : IF TN=0 THEN TN=1 : RETURN ELSE GOSUB 2490 : RETURN
2380 'exec TX$(i)
2390    TXW$="" : FOR I=1 TO TN-1
2400                 IF LEFT$(TX$(I),1)<>"'" THEN TXW$=TXW$+"("+TX$(I)+")"
2410            NEXT
2420    TIME$="00:00:00" : PSW=1 : GOSUB 1010
2430    PRINT TIME$       : PSW=0 : RETURN
2440 '
2450 'mode set
2460    POKE TT,&HFF : PRINT "--- trace mode"   : RETURN
2480 'list
2490    PRINT : IF TN=1 THEN RETURN
2500    FOR I=1 TO TN-1 : PRINT FNTN$(I)+TX$(I) : NEXT : RETURN
2510 'list to printer
2520    OPEN "O",#2,"LPT0:" : PRINT #2
2530    FOR I=1 TO TN-1 : PRINT #2, FNTN$(I)+TX$(I) : NEXT : CLOSE #2 : RETURN
2540 '* get file name subroutine
2550    IF IN$="" THEN 2640
2560    IF LEFT$(IN$,1)=" " THEN IN$=MID$(IN$,2) : GOTO 2550
2570      FOR I=1 TO LEN(IN$) : D1$=MID$(IN$,I,1)
2580      IF INSTR(FLNG$,D1$)<>0 THEN I=255 : NEXT : GOTO 2630
2590      IF D1$<>" " THEN NEXT : FL$=IN$ : IN$="" : GOTO 2610
2600         FL$=LEFT$(IN$,I-1) : IN$=MID$(IN$,I+1) : I=255 : NEXT
2610    IF LEFT$(FL$,4)="TOTO" THEN 2630
```

リスト1　TOTO CCP BASICプログラム・リスト

```
2620     IF LEN(FL$)<9 THEN RETURN
2630     PRINT "*** file name error ;";
2640     INPUT "input file name ---", IN$ : IF IN$="" THEN 2660 ELSE 2560
2650 '
2660     PRINT "*** command ignored" : RETURN 2010
2670 '* error interrupt
2680     IF ERR=63 THEN RESUME 2850
2690     IF ERR<>64 THEN ON ERROR GOTO 0 : RESUME
2700        PRINT "are you sure(Y or N)? ";
2710        SURE$=INKEY$ : IF SURE$="" THEN 2710 ELSE PRINT SURE$
2720                       IF SURE$="Y" THEN KILL FL$ : RESUME
2730                       IF SURE$="N" THEN RESUME 2660 ELSE 2700
2740 'save
2750     IN$=MID$(IN$,6) : GOSUB 2550
2760     ON ERROR GOTO 2680 : HD$="'--- "+FL$+" ---" : FL$=DV$+FL$
2770     OPEN "O",#1,FL$ : IF LEFT$(TX$(1),1)<>"'" THEN PRINT #1,HD$
2780     FOR I=1 TO TN-1 : PRINT #1,TX$(I) : NEXT : CLOSE #1 : RETURN
2790 'load
2800     IN$=MID$(IN$,6) : GOSUB 2550
2810 '(load exec)
2820     ON ERROR GOTO 2680 : LDNG=0 : FL$=DV$+FL$ : OPEN "I",#1,FL$ : I=1
2830     LINEINPUT #1,TX$(I) : IF EOF(1)=0 THEN I=I+1 : GOTO 2830
2840     CLOSE #1 : TN=I+1 : GOSUB 2490 : RETURN
2850     PRINT "*** file "+MID$(FL$,3)+" not found" : LDNG=-1 : RETURN
2860 'run
2870     IN$=MID$(IN$,5)
2880     WHILE IN$<>""
2890       GOSUB 2550 : GOSUB 2820 : IF LDNG=0 THEN GOSUB 2390
2900     WEND : RETURN
2910 'submit
2920     IN$=MID$(IN$,8) : IF IN$="" THEN 2950 ' in case of no file name
2930     GOSUB 2550 : GOSUB 2820 : EX$=""
2940     IF LDNG=-1 THEN PRINT "*** submit command aborted " : RETURN
2950     FOR I=1 TO TN-1 : EX$=EX$+TX$(I)
2960       IF LEFT$(TX$(I),1)<>"'" THEN PRINT "** submit error: "+TX$ : RETURN
2970     NEXT : EX$=MID$(EX$,2)+"'"
2980     WHILE EX$<>""
2990       I=INSTR(EX$,"'"): IN$=LEFT$(EX$,I-1): EX$=MID$(EX$,I+1) : GOSUB 2040
3000     WEND : RETURN
3010 'file dump to printer
3020     IN$=MID$(IN$,6)
3030     WHILE IN$<>""
3040       GOSUB 2550 : GOSUB 2820 : IF LDNG=0 THEN GOSUB 2510
3050     WEND : RETURN
3060 END
```

リスト2　TOTOインタープリタ Kコンパイラ ソース・リスト

```
1000 ' (* y - TOTO  for  FM       ( V-3.6 )  ---- K compiler V-1.3          *)
1010 ' (*                        (C) H.KOMATSU      84.01.04  no-1 TOTOK.A2 *)
1020 'CONST     STOP=$6C26 , NSTE=$6C22
1030 'CONST     NIL=999 , TCMD=998
1040 'CONST     BKFLAG=$6C02 ; KBFLAG=$6C03
1050 'entry  ;  CODE $10,$FF,$6C,$20 ; (* save   S-register   *)
1060 '          POKE KBFLAG,$00 ; A=$1E1 ; SEB=A(0)
1070 '          A=NSTE ; CPOKE A,$7C,STOP#,$3B00#
1080 '                IF PEEK(BKFLAG)=0 THEN GOTO init  ; FI
1090 '                IF PEEK(BKFLAG)=3 THEN GOTO main3 ; FI
1100 '                IF PEEK(BKFLAG)=4 THEN GOTO main4 ; FI
1110 'rtn    ;  A=$1E1 ; A(0)=SEB
1120 '          CODE $35,$FF ;            (* PULS *)
1130 'rtnerr;   A=$1E1 ; A(0)=SEB
1140 '          POKE KBFLAG,KBF
1150 '          IF KBF=1 THEN PRINT "--- cell overflow ---     ",/ ; FI
1160 '      (*  IF KBF=2 THEN PRINT "--- stop key ---          ",/ ; FI  *)
1170 '          IF KBF=3 THEN PRINT "--- stack overflow ---    ",/ ; FI
1180 '          IF KBF=4 THEN PRINT "--- undefined FN = ", CHR$(FNC), / ; FI
1190 '          IF KBF=5 THEN PRINT "--- check '(' and ')' ---",/ ; FI
1200 '          IF KBF=6 THEN PRINT "--- Acc overflow ---      ",/ ; FI
1210 '      (*  IF KBF=C THEN PRINT "--- clear screen ---      ",/ ; FI *)
1220 '          CODE $10,$FE,$6C,$20 ; (* LDS  *)
1230 '          CODE $35,$FF ;           (* PULS *)
1240 '
1250 '(*** system initializer ***)
1260 'init   ;   TTA=$6C04 ; CAR=$3000 ; CDR=$3800 ; STK=$4800
1270 '           FLP=1 ; ATP=NIL ; genATMi[NIL,NIL]
1280 '           initP["T+-FLRAJ#%<>HD*C$NEWS."] ; ATPi=ATP
1290 '           LINE [0,0,639,199,0,1,1]         (* PSET, blue, box *)
1300 '           ACC=0 ; DIR=0 ; CLR=7 ; X=320 ; Y=100
1310 '           GOTO rtn
1320 '
1330 'initP ;   I=%1 ; WHILE I:0)<>'.' ; genATMi[NIL,I:0)] ; I=I+1 ; WEND
1340 '          RETURN
1350 '
1360 '(** read command statement **)
1370 'reads ;   TXP=STXA ; readd[] ; lpar[] ; RETURN
1380 'lpar  ;   readd[] ; IF DL=')' THEN RS=NIL ; RETURN ; FI
1390 '                     IF DL='(' THEN lpar[] ; FI
```

リスト2　TOTOインタープリタ　Kコンパイラ　ソース・リスト

```
1400 '                 push[RS] ;  lpar[] ; pull[] ; cons[] ; RETURN
1410 '
1420 'readd ;  WHILE TXP:0)=$20 ; TXP=TXP+1 ; WEND ; DL=TXP:0) ; TXP=TXP+1
1430 '          IF DL='(' OR DL=')' THEN RETURN ; FI
1440 '          IF DL<'0' OR DL>'9' THEN genATM[NIL,DL] ; DL=0 ; RETURN ; FI
1450 '             NUM=DL-48 ; WHILE TXP:0)>='0' AND TXP:0)<='9'
1460 '                         NUM=NUM*10+(TXP:0)-48) ; TXP=TXP+1
1470 '                WEND ; genATM[-1,NUM] ; DL=0 ; RETURN
1480 '
1490 'genATM ; IF %1=NIL THEN       (* name atom *)
1500 '          FOR I=ATP+1 TO NIL
1510 '             IF CAR(I)>=0 AND CDR(I)=%2 THEN RS=I ; RETURN ; FI
1520 '          NEXT
1530 '         ELSE                 (* num  atom *)
1540 '          FOR I=ATP+1 TO ATPi
1550 '             IF CAR(I)=-1 AND CDR(I)=%2 THEN RS=I ; RETURN ; FI
1560 '          NEXT
1570 '         FI
1580 'genATMi; RS=ATP ; CAR(ATP)=%1 ; CDR(ATP)=%2 ; ATP=ATP-1 ; RETURN
1590 '
1600 '(* read subroutines *)
1610 'push   ; STK=STK-2 ; IF STK>$4000 THEN STK(0)=%1 ; RETURN ; FI
1620 '          KBF=3 ; GOTO rtnerr
1630 'pull   ; LP=STK(0) ; STK=STK+2 ; LP=LP ; RETURN
1640 'cons   ; CAR(FLP)=LP ; CDR(FLP)=RS ; RS=FLP ; FLP=FLP+1
1650 '         IF ATP>FLP THEN RETURN ; FI
1660 '          KBF=1 ; GOTO rtnerr
1670 '
1680 '(** get text pointer from BASIC & check parenthesis **)
1690 'load  ;  TXA=$4000 ; I=TXA(0) ; DTL=I:0) ; I=I+1 ; STXA=I(0)
1700 '         PAR=0 ; TXP=STXA
1710 '         WHILE DTL>0 ; DTL=DTL-1
1720 '           IF TXP:0)='(' THEN PAR=PAR+1 ; ELSE
1730 '           IF TXP:0)=')' THEN PAR=PAR-1 ; FI ; FI
1740 '              TXP=TXP+1
1750 '         WEND
1760 '         IF PAR=0 THEN RETURN ; ELSE KBF=5 ; GOTO rtnerr ; FI
1770 '
1780 '(***  main program  ***)
1790 'main4 ; FLP=1 ; ATP=ATPi
1800 'main3 ; TT=TTA:0) ; TCNT=0 ; W8=0 ; NEWS=0
1810 '        A=$1E1;A(0)=NSTE; POKE STOP,0 ; (* stop key *)
1820 '        load[] ; reads[] ; eval[1,RS] ; GOTO rtn
1830 '
1840 'eval  ;  (* %1=n , %2=s *)
1850 '        IF %2=NIL   THEN RETURN ; FI
1860 '        IF PEEK(STOP)<>0 THEN KBF=2 ; GOTO rtnerr ; FI
1870 '           cars=CAR(%2)
1880 '        IF cars=TCMD THEN test[CDR(%2)] ; RETURN ; FI
1890 '        IF CAR(cars)=NIL THEN GOTO apply ; FI (*[%1,cars]*)
1900 '           caars=CAR(cars)
1910 '        IF caars<0   THEN %1=CDR(cars) ; %2=CDR(%2) ; GOTO eval ;FI
1920 '        IF caars<ATP THEN evloop[%1,caars] ; GOTO evcdr ; FI
1930 '        IF cars<ATP  THEN evloop[%1,cars]  ; GOTO evcdr ; FI
1940 '
1950 'evcdr ; %1=1   ; %2=CDR(%2) ; GOTO eval
1960 'accum ; %1=ACC ; %2=CDR(%2) ; GOTO eval
1970 '
1980 'evloop; WHILE %1>0 ; %1=%1-1 ; eval[1,%2] ; WEND ; RETURN
1990 '
2000 'dfine ; cadrs=CAR(CDR(%2)) ; cddrs=CDR(CDR(%2))
2010 '        IF   CAR(cddrs)>ATP THEN
2020 '              CAR(cadrs)=   cddrs ; RETURN
2030 '        ELSE CAR(cadrs)=CAR(cddrs); eval[1,CDR(cddrs)] ; RETURN ; FI
2040 '
2050 'test  ; IF TT<>0 THEN traceT[] ; FI
2060 '        IF ACC>0 THEN eval[1,CAR(    %1 )]
2070 '                 ELSE eval[1,CAR(CDR(%1))] ; FI
2080 '                      eval[1,CDR(CDR(%1))] ; RETURN
2090 '
2100 'apply ;     FNC=LOW(CDR(cars))
2110 '         IF TT<>0    THEN tracef[%1,FNC] ; FI
2120 '         IF FNC='+'  THEN GOTO plus      ; FI
2130 '         IF FNC='-'  THEN GOTO minus     ; FI
2140 '         IF FNC='F'  THEN GOTO forward   ; FI
2150 '         IF FNC='L'  THEN GOTO left      ; FI
2160 '         IF FNC='R'  THEN GOTO right     ; FI
2170 '         IF FNC='A'  THEN GOTO accum     ; FI
2180 '         IF FNC='J'  THEN GOTO jump      ; FI
2190 '         IF FNC='#'  THEN GOTO iro       ; FI
2200 '         IF FNC='%'  THEN GOTO percent   ; FI
2210 '         IF FNC='<'  THEN GOTO pushs     ; FI
2220 '         IF FNC='>'  THEN GOTO pulls     ; FI
2230 '         IF FNC='H'  THEN GOTO home      ; FI
2240 '         IF FNC='D'  THEN GOTO dfine     ; FI
2250 '         IF FNC='*'  THEN GOTO way8      ; FI
2260 '         IF FNC='C'  THEN GOTO clear     ; FI
2270 '         IF FNC='$'  THEN GOTO paintc    ; FI
2280 '         IF FNC='N'  THEN GOTO north     ; FI
```

リスト2　TOTOインタープリタ Kコンパイラ ソース・リスト

```
2290 '          IF NEWS=0   THEN GOTO skip        ; FI
2300 '          IF FNC='E'  THEN GOTO east        ; FI
2310 '          IF FNC='W'  THEN GOTO west        ; FI
2320 '          IF FNC='S'  THEN GOTO south       ; FI
2330 'skip    ; IF cars=NIL THEN GOTO evcdr       ; FI
2340 '            KBF=4 ; GOTO rtnerr
2350 'plus    ; ACC=ACC+%1 ;                           traceA[] ; GOTO evcdr
2360 'minus   ; ACC=ACC-%1 ; IF ACC<0 THEN ACC=0 ; FI ; traceA[] ; GOTO evcdr
2370 'percent; ACC=ACC*%1/100 ; IF ACC>=0 THEN traceA[] ; GOTO evcdr ; FI
2380 '                             KBF=6 ; GOTO rtnerr
2390 'forward; X0=X ; Y0=Y
2400 '          STEP=%1 ; newXY[] ;  (* %1 --> X, Y *)
2410 '             LINE [X0,Y0,X,Y,0,CLR,0] ; (* PSET, color, normal *)
2420 '             GOTO evcdr
2430 'jump    ; STEP=%1 ; newXY[] ; GOTO evcdr
2440 'right   ; DIR=DIR+%1 ; WHILE DIR>=360 ; DIR=DIR-360 ; WEND ; GOTO evcdr
2450 'left    ; DIR=DIR-%1 ; WHILE DIR<0    ; DIR=DIR+360 ; WEND ; GOTO evcdr
2460 'pushs   ; push[ACC]; push[DIR]; push[CLR]; push[X]; push[Y]; GOTO evcdr
2470 'pulls   ; Y=pull[] ; X=pull[] ; CLR=pull[] ; DIR=pull[] ; ACC=pull[] ;
                 GOTO evcdr
2480 'clear   ; CLS[0,7,0]
2490 '             LINE [0,0,639,199,0,1,1] ; (* PSET, blue , box *)
2500 '             POKE KBFLAG,$43 ; (* "C" *)
2510 '          FUNC=LOW(CDR(CAR(CDR(%2))))        (* next command *)
2520 '          IF FUNC='C' THEN NEWS=1 ; %2=CDR(%2) ; FI
2530 '             GOTO evcdr
2540 'home    ; X=320 ; Y=100 ; GOTO evcdr
2550 'north   ; DIR=0         ; GOTO evcdr
2560 'east    ; DIR=90        ; GOTO evcdr
2570 'west    ; DIR=270       ; GOTO evcdr
2580 'south   ; DIR=180       ; GOTO evcdr
2590 'iro     ; CLR=%1; WHILE CLR>7; CLR=CLR-7; WEND ; COLOR[CLR] ; GOTO evcdr
2600 'way8    ; W8=$0F ; GOTO evcdr
2610 'paintc  ; PAINT[X,Y,%1,3,CLR,%1,1] ; GOTO evcdr
2620 '
2630 '(* subroutines of apply *)
2640 'tracef  ; IF %1<>1 THEN PRINT %1 ; FI ; PRINT CHR$(LOW(%2))," "
2650 '             TCNT=TCNT+1 ; IF TCNT>150 THEN TT=0 ; FI ; RETURN
2660 'traceT  ; PRINT "T "
2670 '             TCNT=TCNT+1 ; IF TCNT>150 THEN TT=0 ; FI ; RETURN
2680 'traceA  ; IF TT=0  THEN RETURN   ; FI ; PRINT "(A=",ACC,") "   ; RETURN
2690 '
2700 'CLS     ; HLT[] ; CPOKE ADDR,$02,%1,%3,%2,%2 ; SUB[] ; RETURN
2710 'LINE    ; HLT[] ; CPOKE ADDR,$15,%6,%5,%1#,%2#,%3#,%4#,%7 ; SUB[] ; RETURN
2720 'COLOR   ; HLT[] ; CPOKE ADDR,3,2,$11,%1 ; SUB[] ; RETURN
2730 'PAINT   ; HLT[] ; CPOKE ADDR,$18,%1#,%2#,%3,%4,%5,%6,%7 ; SUB[] ; RETURN
2740 ' HLT    ; CODE $B6,$FD05#,$2BFB# ; POKE $FD05,$80
2750 '          CODE $B6,$FD05#,$2AFB# ; ADDR=$FC82 ; RETURN
2760 ' SUB    ; POKE $FD05,0 ; RETURN
2770 '
2780 'newXY   ; IF W8=0 THEN
2790 '            XF=sin[DIR]*STEP/100*155/70 ; X=X+XF
2800 '            YF=cos[DIR]*STEP/100        ; Y=Y-YF ; RETURN ; FI
2810 'genDIR ;
2820 '          IF DIR< 22 THEN          Ydec[] ; RETURN ; FI
2830 '          IF DIR< 67 THEN Xinc[] ; Ydec[] ; RETURN ; FI
2840 '          IF DIR<112 THEN Xinc[]          ; RETURN ; FI
2850 '          IF DIR<157 THEN Xinc[] ; Yinc[] ; RETURN ; FI
2860 '          IF DIR<202 THEN          Yinc[] ; RETURN ; FI
2870 '          IF DIR<247 THEN Xdec[] ; Yinc[] ; RETURN ; FI
2880 '          IF DIR<292 THEN Xdec[]          ; RETURN ; FI
2890 '          IF DIR<337 THEN Xdec[] ; Ydec[] ; RETURN
2900 '                     ELSE          Ydec[] ; RETURN ; FI
2910 'Xinc    ; X=X+STEP+STEP ; RETURN
2920 'Xdec    ; X=X-STEP-STEP ; RETURN
2930 'Yinc    ; Y=Y+STEP      ; RETURN
2940 'Ydec    ; Y=Y-STEP      ; RETURN
2950 '
2960 'cos     ; %1=90-%1
2970 'sin     ; WHILE %1<0   ; %1=%1+360 ; WEND
2980 '          WHILE %1>360 ; %1=%1-360 ; WEND
2990 '          IF %1>270 THEN Z=360-%1 ; ELSE
3000 '          IF %1>180 THEN Z=%1-180 ; ELSE
3010 '          IF %1> 90 THEN Z=180-%1 ; ELSE Z=%1 ; FI ; FI ; FI
3020 '            Z=sintbl[Z] ; IF %1>180 THEN Z=-Z ; FI ; Z=Z ; RETURN
3030 '
3040 'sintbl ; CODE $30,$8C05#,$EC62#,$E68B#,$39
3050 '          CODE  0,  2,  3,  5,  7,  9, 10, 12, 14, 17
3060 '          CODE 17, 19, 21, 23, 24, 26, 28, 29, 31, 33
3070 '          CODE 34, 36, 37, 39, 41, 42, 44, 45, 47, 48
3080 '          CODE 50, 52, 53, 54, 56, 57, 58, 60, 62, 63
3090 '          CODE 64, 66, 67, 68, 69, 71, 72, 73, 74, 75
3100 '          CODE 77, 78, 79, 80, 81, 82, 83, 84, 85, 86
3110 '          CODE 87, 87, 88, 89, 90, 91, 91, 92, 92, 93
3120 '          CODE 94, 95, 95, 96, 96, 97, 97, 97, 98, 98
3130 '          CODE 98, 99, 99, 99, 99,100,100,100,100,100
3140 '          CODE 100
3150 'END
```

ディスクコミュニケーション DISK COMMUNICATION

■ポパイ&ジェリー

念願のミニフロッピーディスクを買いました．ディスクを使っているうちに，カセットのときと違ってLOADするときはファイル名が省略できないのが不便なので，このプログラムを作ってみました．

使い方

まず記録してあるディスケットを０ドライブに入れてRUNしてください．

ファイル名に番号がついて横５個ずつ並びます．ファイル名は属性によって，次のように色分けして表示されます．

BASICプログラム・ファイル	→白色
データ・ファイル	→黄色
マシン語プログラム・ファイル	→水色
ASCII形式	→緑色

そのままファイル名についている番号を入力するとディスクからプログラムを読み込んで自動的にRUNします．

０を入力するとメニュー（表１）が出ます．

表１　メニュー

0	ラン	ディスクからプログラムを読みこんで自動的に走ります
1	ロード	LOADと同じです(BASIC,マシン語は自動的に判断します)
2	ドライブ	ディスクのドライブ番号の変更
3	ネーム	NAMEと同じです．変更する番号と新しいファイル名を入力してください
4	Kill	消したいプログラムの番号を入力してください
5	アドレス	マシン語のSTART番地，END番地，EXEC番地を捜します
6	Fリスト	ファイル名をプリンタに出力します(マシン語はアドレスも出力します)

メニュー６のFリストはマシン語が多いとアドレスを捜すために少々時間がかかります．

なお，FM-7の方はシステム・ディスクの中にある"AUTOUTY"を使って自動的にこのプログラムが走るようにすればディスクもいっそう使いやすくなると思います．そのときはこのプログラムをファイル名"STARTUP"としてセーブしてください．また，PF1にRUN"st"と登録してあるので100行の**KEY1, "R."+CHR$(34)+"st"+CHR$(13)**の"st"を"STARTUP"に直してください。

ディスク・コミュニケーション プログラム・リスト

```
10 REM-----      Kawada & Ozaki  DISK ｺﾐｭﾆｹｰｼｮﾝ     -----
100 CLEAR1500:DEFINTA-V:DIMF$(152):DIMT(152):DEFFNX=ASC(INPUT$(1,1))*256+ASC(INP
UT$(1,1)):KEY1,"R."+CHR$(34)+"st"+CHR$(13):K$=CHR$(14):E$=K$+STRING$(40,35):G$=K
$+"#"+SPACE$(38)+"#":ONERRORGOTO320
110 WIDTH80,25:CONSOLE0,25,0,0:Z=1:T=1:C=4:FORA=1TO19:GOSUB240:GOSUB250:C=C+1:NE
XTA
120 B$="RUN"
130 GOSUB220:IFFC=0 THEN 210
140 IF B$="LOAD" THEN 160
150 ON T(FC)+1 GOTO 170,170,180
160 ON T(FC)+1 GOTO 190,190,200
170 LOAD F$,R
180 LOADM F$,,R
190 LOAD F$
200 LOADM F$:END
210 LOCATE0,23:PRINT"0.ﾗﾝ  1.ﾛｰﾄﾞ  2.ﾄﾞﾗｲﾌﾞ  3.ﾈｰﾑ  4.Kill  5.ｱﾄﾞﾚｽ  6.Fﾘｽﾄ":ONI
NSTR("0123456",INPUT$(1))GOTO120,370,330,350,360,340,390:BEEP:GOTO 210
220 GOSUB410:PRINTB$;" No.";:INPUTFC:IF FC>Z-1 OR FC<0THEN210
230 F$=D$+F$(FC):RETURN
240 L$=DSKI$(D,T,C):V=0:AD=PEEK(VARPTR(L$)+1)*256+PEEK(VARPTR(L$)+2):RETURN
250 F$=MID$(L$,(V)*32+1,16):B$=LEFT$(F$,1):IFB$=CHR$(0)THEN300
260 IFB$=CHR$(255)THEN310
270 A$=LEFT$(F$,8):T(Z)=PEEK(AD+32*V+11):S=PEEK(AD+32*V+12):F$(Z)=A$
280 W=7-T(Z):IF W=7ANDS=255THENW=4
290 COLOR7:PRINTRIGHT$(STR$(Z),2);"  ";:COLORW:PRINTA$,:Z=Z+1
300 IFV=7THENRETURNELSEV=V+1:GOTO250
```

ディスク・コミュニケーション　プログラム・リスト

```
310 PRINT:PRINT:COLOR7:PRINTDSKF(D);"Clusters Free":LOCATE0,22:COLOR12:PRINT"ｱｽｷ
ｰ",:COLOR13:PRINT"ｷｶｲｺﾞ",:COLOR14:PRINT"ﾃﾞｰﾀ",:COLOR15:PRINT"Basic",:COLOR7:PRIN
T"[0･ﾒﾆｭｰ]":GOTO120
320 BEEP:RESUME 100
330 B$="ﾄﾞﾗｲﾌﾞ":GOSUB220:IF FC>4 THEN 330 ELSE D=FC:D$=RIGHT$(STR$(FC),1)+":":GO
TO110
340 B$="ｱﾄﾞﾚｽ":GOSUB220:J=FC:GOSUB420:LOCATE0,21:PRINTF$,X$;Y$;Z$:GOTO120
350 B$="ﾁｪﾝｼﾞﾈｰﾑ":GOSUB220:LOCATE0,23:PRINTF$,:INPUT"ﾆｭｰﾈｰﾑ";N$:GOSUB380:NAME F$
 AS D$+N$:GOTO110
360 B$="Kill":GOSUB220:LOCATE0,23:PRINTF$:GOSUB380:KILLF$:GOTO110
370 B$="LOAD":GOTO130
380 SYMBOL(300,185),"Are You sure(y/n)",1,1:A$=INPUT$(1):IFA$="Y"ORA$="y"THENRET
URNELSE120
390 GOSUB410:INPUT"ﾀｲﾄﾙ";X$:GOSUB410:PRINT"Wait":OPEN"O",1,"LPT0:":PRINT#1,E$:PR
INT#1,G$:PRINT#1,K$+"#       "+X$+SPACE$(33-LEN(X$))+"#":PRINT#1,G$:PRINT#1,E$:PRI
NT#1,CHR$(10);CHR$(10)+CHR$(15):CLOSE:FORJ=1TOZ-1:IFT(J)=2THENGOSUB420ELSEX$="":
Y$="":Z$=""
400 OPEN"O",1,"LPT0:":PRINT#1,K$;F$(J),T(J);X$;Y$;Z$:CLOSE:NEXT:GOTO130
410 LOCATE 0,23:LINE(0,184)-(639,199),PRESET,,BF:RETURN
420 OPEN"I",1,D$+F$(J):Y$=INPUT$(1,1):X=FNX:Y=FNX:X$="START:&H"+HEX$(Y):Y$=" END
:&H"+HEX$(X+Y-1):FORI=1TOX:R$=INPUT$(1,1):NEXT:R$=INPUT$(3,1):Z$=" EXEC:&H"+HEX$
(FNX):CLOSE:X=0:RETURN
```

RANDOM BOX

FM-7 FILES命令の改良

■カメ

FMシリーズのFILES命令は画面サイズを無視したようにどんどんスクロールしてしまい，見づらくてしかたありません．いままでに何本かFILES命令を改良するソフトが発表されましたが，どのディスケットでも使えるものがありませんでした．というのも，同じFMの中でもDISK-BASICに何種類かのバージョンがあるので，作者と同じバージョンを持っていない人は書き変えても暴走してしまうのです．

そこでこのプログラムは**FILES**命令を改良することはもち論，どんなDISK-BASICのバージョンでも対応し，さらにFM-7でもFM-8でも動くスーパープログラムです．

プログラムを$2000～$209Bまで打ち込んで，**SAVEM "FILES"，&H2000，&H209B，&H2000**でセーブしておいてください．次に改良したいディスケットをドライブ0に入れ，**EXEC&H2000** RETURN で，後はすべてプログラムがやってくれます．もしエラーがあると**BEEP**音が鳴り，正常ならすぐ**READY**がでます．リセット・ボタンを押して**FILES**を実行してみてください．**WIDTH40**なら2つずつ，**WIDTH80**なら横4つずつ出力されます．

本プログラムはDISK-BASICを書き変えるだけなのでフリーエリアの減少はありません(**CLEAR**命令は$71D5までOK)．使用中のディスケットも改良できますが，セクタ単位のリード・ライトをやっているのでまず壊れてもよいディスケットでテストしてみてください．改良するディスケットはシステム・ディスクの"SYSDSK"でFORMAT，DISK-BASICをコピーしたものに限ります．

アセンブル・リスト

```
0001  0000                   ;   FILES instruction
0002  0000                   ;   DISK BASIC CONVERT
0003  0000                   ;   22,DEC.1983 FM-7/8
0004  0000                   ;     by S.NAGAKAWA
0005  0000                   ;
0006  2000                         ORG   $2000
0007  2000 CC0A8E                  LDD   #$0A8E
0008  2003 CE2078                  LDU   #DATA
0009  2006 B72094                  STA   RCB
0010  2009 F18000                  CMPB  $8000
0011  200C 2610                    BNE   FM-8
0012  200E CC9C22                  LDD   #$9C22
0013  2011 ED4D                    STD   13,U
0014  2013 EDC810                  STD   16,U
0015  2016 EDC819                  STD   25,U
0016  2019 CCB615                  LDD   #$B615
0017  201C ED4A                    STD   10,U
0018  201E 8D3A             FM-8:  BSR   DISK
0019  2020 6D01                    TST   1,X
0020  2022 264E                    BNE   ERROR
0021  2024 8E2100                  LDX   #$2100
0022  2027 861F                    LDA   #$1F
0023  2029 A180             LOOP:  CMPA  ,X+
0024  202B 2707                    BEQ   ENTRY
0025  202D 8C2200                  CMPX  #$2200
0026  2030 2440                    BCC   ERROR
0027  2032 20F5                    BRA   LOOP
0028  2034                  ;
0029  2034 ECC1             ENTRY: LDD   ,U++
0030  2036 ED01                    STD   1,X
0031  2038 ECC1                    LDD   ,U++
0032  203A ED03                    STD   3,X
0033  203C ECC1                    LDD   ,U++
0034  203E ED06                    STD   6,X
0035  2040 8E22E0                  LDX   #$22E0
0036  2043 860B                    LDA   #11
0037  2045 10AEC1           LOOP2: LDY   ,U++
0038  2048 10AF81                  STY   ,X++
0039  204B 4A                      DECA
0040  204C 26F7                    BNE   LOOP2
0041  204E 8609                    LDA   #9
0042  2050 B72094                  STA   RCB
0043  2053 8D05                    BSR   DISK
0044  2055 6D01                    TST   1,X
0045  2057 2619                    BNE   ERROR
0046  2059 39                      RTS
0047  205A                  ;
0048  205A 8E2094           DISK:  LDX   #RCB
0049  205D CC210C                  LDD   #$210C
0050  2060 A702                    STA   2,X
0051  2062 E705                    STB   5,X
0052  2064 BD00DE                  JSR   $00DE
0053  2067 CC2210                  LDD   #$2210
0054  206A A702                    STA   2,X
0055  206C E705                    STB   5,X
0056  206E BD00DE                  JSR   $00DE
0057  2071 39                      RTS
0058  2072                  ;
0059  2072 8641             ERROR: LDA   #$41
0060  2074 B7FD03                  STA   $FD03
0061  2077 39                      RTS
0062  2078                  ;
0063  2078 12121212         DATA:  FDB   $1212,$1212
0064  207C 7FE03404                FDB   $7FE0,$3404
0065  2080 4FBDB96D                FDB   $4FBD,$B96D
0066  2084 BDA026BD                FDB   $BDA0,$26BD
0067  2088 A0263504                FDB   $A026,$3504
0068  208C C10A2403                FDB   $C10A,$2403
0069  2090 7EA02639                FDB   $7EA0,$2639
0070  2094                  ;
0071  2094 0A002100         RCB:   FDB   $0A00,$2100
0072  2098 000C0100                FDB   $000C,$0100
```

FM-7/11

拡張Kコンパイラ V3.6

リカーシブコールのできるコンパイラ

■COMPAC　Td & St & Sgn

K言語は，パソコン用コンパイラ言語として考案されたもので，その文法は比較的BASICに近く，機能を縮小してコンパイラ自体を小さくできるように配慮されています．

拡張Kコンパイラは，そのK言語にいくつかのコマンドを加え，さらに6809CPUの特徴を生かすべく設計されたコンパイラで，次のような特徴があります．

①引数を使ったサブルーチン・コールや，リカシーブル・コールができ，サブルーチンへの引数は値渡しで行なえる．
②行番号の概念はなく，英数字のラベルが使える．
③IF-THEN-ELSEの複数行化ができ，REPEAT-UNTILも使える．
④REPEAT, WHILEのループ中への飛び込み，飛び出しができる．
⑤マシン語プログラムをプログラム中に書ける．
⑥コンパイラ自体とコンパイル後のオブジェクト（マシン語プログラム）は，ポジションインデペンデント（位置自由）である．
⑦ランタイム・ルーチン（位置固定）が独立しており，機種に依存するのは１文字入出力と１行入力の部分だけである．

拡張Kコンパイラの使い方（オンメモリ版）

コンパイラの作り方

リスト５のダンプ・リストを打ち込み，ファイル名〝RT〟でセーブします．次に，**リスト６**のダンプ・リストを打ち込み，ファイル名〝K〟でセーブします．FM-11の方は，**リスト７**に従い，ファイル版に変更を参照してください．

コンパイル時には，〝RT〟と〝K〟が必要ですが，実行時には〝RT〟があれば実行できます．

ソース・プログラムの作り方

拡張K言語のソース・テキスト（プログラム）はF-BASICの**REM**文の形で作成します．したがって，１行の文の先頭には必ず『'』がなくてはいけません．

リスト１のプログラム例は，１から10の範囲の乱数を100個とり，偶数と奇数のどちらが多いかをみるプログラムです．行番号はテキスト作成を行なう場合は必要なく，25行にある**GOTO**文のジャンプ先は，20行に定義されているラベル『GETRAND』です．また，BASICでは１行に複数のステートメントを記述する場合，その区切りは『:（コロン）』で行ないますが，拡張Kでは『;（セミコロン）』を使います．ですから，25行にある『;』は，**PRINT**文の出力を連続させるためのものではなく，次の『**fi**』との区切りを意味します．拡張Kの**PRINT**文は指定（改行子『/』を書く）をしないかぎり，改行は行ないません．

拡張Kでは英字の大小文字は区別しないので（文字列中は除く），キーワードを小文字，その他を大文字として，リストを見やすくすることもできます．その他，詳しい文法などはここでは省略しますが，拡張Kを初めて使う方は，まず次ページの**リスト１**を入力してみてください．入力したプログラムは，通常のBASICと同様にテープにセーブできます．

コンパイル

ソース・テキストの入力後，**EXEC&H2000**⏎と入力すると，コンパイルが開始されます．コンパイル・エラーが生じた場合，その行と位置を表示するので，ソース・テキストを修正して，再度コンパイルを行なってください．

コンパイルが正常に終了すると，オブジェクト・プログラムのサイズが表示されます．これはオブジェクトをセーブする場合に必要となるので，値をメモなどに記入しておいてください．**リスト１**のプログラムをコンパイルすると，図１のようになります．

実　行

コンパイルしたオブジェクトは$5000より生成されるので，**EXEC &H5000**⏎で，すぐに実行することが可能です．しかし，プログラムの作りによっては暴走する場合もあるので，実行前にソース・テキストをテープなどにセーブしておいてください．

リスト1のプログラムを実行すると，図2のようになります．**リスト２**は**リスト１**と同様のプログラムを，BASICで記述したものです．入力してその実行速度を比較すると，Kコンパイラ・オブジェクトの実行の速さがわかるでしょう．

完成したオブジェクトは，そのままマシン語プログラムとして，セーブすることができます．このとき，ランタイム・ルーチンも合わせてセーブすれば，再び実行するときにランタイム・ルーチンを別にロードする必要がなくなります．セーブするコマンドは通常のマシン語と同様，

リスト1　サンプル・プログラム

```
10 '(* ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ *) KISUU=0; GUUSU=0
15 'for N=1 to 100
20 ' GETRAND; RD=rnd(11)
25 ' if RD=0 then goto GETRAND else print "."; fi
30 ' if (RD and $0001)=0 then GUUSU=GUUSU+1
35 '                    else KISUU=KISUU+1
40 ' fi
45 'next
50 'print /,"ｷｽｳ  : ", KISUU, " ｶｲ",/,"ｸﾞｳｽｳ: ", GUUSU, " ｶｲ",/
55 'end
```

リスト2　比較プログラム(BASIC)

```
10 'ｻﾝﾌﾟﾙ ﾌﾟﾛｸﾞﾗﾑ
15 RANDOMIZE
20 FOR N=1 TO 100
25  RD%=RND*10
30  IF RD%=0 THEN 25 ELSE PRINT ".";
35  IF (RD% AND &H1)=0 THEN GUUSU=GUUSU+1 ELSE KISUU=KISUU+1
40 NEXT
45 PRINT:PRINT "ｷｽｳ  : "; KISUU; " ｶｲ": PRINT "ｸﾞｳｽｳ: "; GUUSU; " ｶｲ"
50 END
```

図1　コンパイル画面

```
Ready
EXEC&H2000

K COMPILER VERSION 3.3
COPYRIGHT 1982 BY COMPAC.


COMPILING...




OBJECT PROGRAM SIZE :179 ($00B3)

COMPILE END.

Ready
```

図3　Kコンパイラ・メモリ・アップ

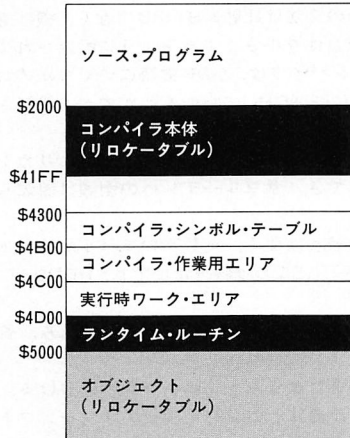



図2　実行画面

```
Ready
EXEC&H5000
................................................................................
....................
ｷｽｳ  : 48 ｶｲ
ｸﾞｳｽｳ: 52 ｶｲ

Ready
HARDC
```

SAVEM "File Name", &H4D00, End Address, &H5000

となり，End Addressはオブジェクト・プログラムのサイズに&H5000を加えたものです．File Nameは任意に付けてください．

文	機　　能
GET	キーボードにより1文字を入力する．キー入力がないときは，0を値として持つ．ランタイム・ルーチンの$4F89を$00から$01に変更すると，キー入力があるまで待つようになる．
INPUT	キーボードから⏎キーが押されるまでを入力とする．DELキー，INSキーで，入力した文字の削除，挿入ができる．CTRL＋Cで実行を中止する．
PRINT	画面に出力する．ランタイム・ルーチンの$4F38を$8Dにすると，画面とプリンタの両方に出力する．プリンタにだけ出力するには，$4F39を$15にする．

オブジェクトは，先頭で**PSHS**，**END**文および文の最後で**PULS**を行なって，スタック・ポインタおよびレジスタ類を全部保護するので，他のルーチン(BASICなど)から，サブルーチンとして利用できます．

裏RAMの使用(FM-7のみ)

FM-7では，BASICインタープリタROMとアドレスが重複しているRAMを，ソフトウェアで切り換えて使うことができます．このRAMエリアを裏RAMと呼び，ここへコンパイラを移動することで，テキスト・エリア，オブジェクト・エリアを増やすことが可能です．

拡張Kコンパイラを裏RAMに移して使うには，次のようにしてください．

(1)通常の手段で，ランタイム，コンパイラ本体をロードします．

(2)**リスト3**のリロケータを入力して，**EXEC&H2000**⏎でコンパイルします．

リスト3　リロケータ

```
10 'PRINT"RELOCATER"/;POKE $FD0F,0
20 'PRINT"START ADR=";SA=INPUT
30 'PRINT"END   ADR=";EA=INPUT
40 'PRINT"DEST  ADR=";DA=INPUT
50 'WHILE SA<=EA;DA:0)=SA:0)
60 '    DA=DA+1;SA=SA+1;WEND
70 'A=PEEK($FD0F)
```

リスト4　エグゼキュータ

```
10 'PRINT"EXECUTOR"/;POKE $FD0F,0
20 'PRINT"START ADR=";SA=INPUT
30 'CALL SA
40 'A=PEEK($FD0F)
```

(3)**EXEC&H5000**⏎でリロケータを実行すると，スタート，エンド，ディスティネーションの各アドレスの入力要求をしてきます．たとえば次のように入力してください（アンダーラインの部分を入力する）．

```
START ADDR=$2000 ⏎ ←Kコンパイラの先頭アドレス
END   ADDR=$41FF ⏎ ←Kコンパイラの最終アドレス
DEST  ADDR=$C000 ⏎ ←Kコンパイラを移す先頭アドレス
```

なお，このリロケータのオブジェクトを，ランタイム・ルーチンと合わせてセーブし，裏RAM用リロケータとすれば，次回から(2)の作業は不要になります．

(4)リロケートが済んだら，**リスト4**のエグゼキュータのリストを入力し，コンパイルします．

(5)コンパイルが済んだらエグゼキュータ・オブジェクトは単独でセーブしておきます．

(6)以上でコンパイルの準備は完了なので，

```
CLEAR100, &H41FF ⏎
```

と入力してから，コンパイルしたいプログラムをロードしてください．通常の拡張Kコンパイラに比べ，テキスト・エリアが増えるので，従来メモリ不足で入力できなかったソース・テキストもかなり入力することができます．

(7)ソース・テキストの入力が済んだら，万一のことを考えてテープなどにセーブしてください．このあと，(5)でセーブしたエグゼキュータをロードし，

```
EXEC&H5000 ⏎
```

で実行します．すると，コンパイラのスタート・アドレスを聞いてくるので，

```
EXEC ADDR=$C000 ⏎
```

（これは(3)の『DEST ADDR』で入力した値）と入力すれば，裏RAMにリロケートしたコンパイラが起動します．コンパイラを起動する前には必ずエグゼキュータをロードしてください．

ファイル版への変更（FM7/11）

通常の拡張Kコンパイラは，メモリ上にあるソース・テキストをコンパイルしますが（これを便宜上オン・メモリ版と呼びます），テキスト・エリアが約6Kとバイト限られているため，長いプログラムのコンパイルはできません．

これに対し，ファイル版の拡張Kコンパイラは，ディスクあるいはテープにASCII形式でセーブしたソース・プログラムをコンパイルするので，オブジェクト・エリアの許す限りのソースをコンパイルすることができます．

オン・メモリ版をファイル版に変更するには，コンパイラ本体とランタイム・ルーチンを，**リスト7**のダンプ・リストに従って変更し，**リスト8**のメイン・プログラムを入力してください．入力後，次のようにしてディスクまたはテープにセーブします．

```
SAVE "MAIN" ⏎
SAVEM "RTF", &H4D00, &H4FFF, &H4D00 ⏎
SAVEM "KF", &H2000, &H41FF, &H2000 ⏎
```

FM-11では$4FACからの２バイトを，$E537に変更してください．これは，F-BASIC ROMの１行入力ルーチンのアドレスです．F-BASICのバージョンによって，変わることがあるので注意してください．また，$4FA1を$2Cに，$4FE2を$39に変更してください．

変更したファイル版は次の順序で使います．

①ソース・テキストを入力し，それを使っている外部記憶装置（ディスクまたはテープ）にASCII形式でセーブします．

②**リスト8**のメイン・プログラムをロードしRUNします．すると，"HOT OR COLD?"とメッセージがでるので，ランタイムおよびコンパイラがメモリ上にロード済みであれば[H]を，そうでなければ，変更したランタイムとコンパイラをセーブしたディスク（ドライブ0）またはテープをセットしてから，[C]をキーインしてください．

③次に，"FILE NAME？"とメッセージがでるので，①でセーブしたソース・テキストのファイルをセットしてからファイル名を入力します．

④ファイル名を入力すると，ソース・テキストをファイルから読みながらコンパイルを始め，できあがったオブジェクトはメモリ版同様に実行，セーブが可能です．

表1　Kコンパイラ　エントリ・マップ（V3.3）

アドレス	内　容	機　　能
2003	4C	コンパイル時変数エリア・アドレスの上位1バイト．
2007,8	4C00	コンパイル時Uスタック・エリアの最終アドレス．
2073,4	4200	プログラム1行入力バッファ・アドレス．
207D,E	5000	オブジェクト作成アドレス．
2082,3	4300	シンボル・テーブル・アドレス．
208E,F	4B8E	コンパイル時スタック・エリアの最終アドレス．
209A,B	4B00	コンパイル時作業用エリア．
20C0	4C	実行時変数エリア・アドレスの上位1バイト．
20CE	4C	実行時Uスタック・エリア最終アドレスの上位1バイト．
20D4	00	実行時Uスタック・エリア最終アドレスの下位1バイト．
4186	サブルーチン	コンパイラ1行入力ルーチン．
41E2	〃	コンパイラOPENルーチン．
41EE	〃	コンパイラCLOSEルーチン．
4D00	〃	ランタイム1文字入力ルーチン．エコーバックなし．
4D03	〃	ランタイム1文字出力ルーチン．
4D06	〃	ランタイム1行入力ルーチン．エコーバックあり．バック・スペースあり．

バージョン・アップ

V3.6では，旧バージョンに次の変更，デバッグをしたので，旧デバージョンをお持ちの方は，**リスト9**に従って変更してください．

❶コンパイル・エラーのとき，暴走することがあった．

対策：コンパイルの最初でSスタックを保存し，最後でSスタックを復帰した後，BASICに戻るようにした．コンパイラをサブルーチンとして使う方は，$41F7を$7Fから$FFに変えれば，レジスタ類が全部保護されて戻ってくるようになる．これに伴ない，旧バージョンの$2001は$2003に，$2005は$2007に変わっている．

❷オブジェクト・プログラムが，サブルーチンとならない．
対策：オブジェクト・プログラムの先頭で，**PSHS**を行ない，Sスタックを保存し，**END**文およびオブジェクト・プログラムの最後で，Sスタックを復帰（**PULS**）して戻るようにした．

これによって，オブジェクト・プログラム内で，たとえSスタックを変更しても，正常に復帰できる．

注）以前のように，強制的にBASICに戻すことはしないので，プログラムでRAMモードにしたときは，最後に必ずROMモードに戻す．

❸論理演算子**AND**を英小文字で使うと，論理和**OR**となった．また，配列を用いて**AND**を3個以上続けると，論理和**OR**となった．
対策：修正．

❹**PEEK**文で変数を使うと，２バイト**PEEK**になった．
対策：修正．

❺**POKE**文，**CPOKE**文の中で，**PEEK**文が動作しなかった．
対策：修正．

❻ラベル・テーブルが小さかった．
対策：ラベル・テーブルを$100ふやした．

注意・その他

配列の使い方

拡張K言語での配列は，１バイト配列と２バイト配列があり，その領域は，実アドレスで指定します．

配列を使うには，そのデータをメモリ上のどこに配置するかを，ユーザー自身で決定し，その先頭アドレスを，

```
変数名＝式（式は配列の先頭アドレス）
```

で指定します．ここで使った変数名が，以後配列名として使われます．

配列要素のアクセスは下記のように行ないますが，要素数の上限を考慮して，配列エリアにはプログラムが書かれないようにし，また配列エリアを超えて書き込みが行なわれないように，注意が必要です．

配列は変数の値を変えることで，その位置を簡単に変更することができるので，プログラム中で配列を動的に処理することも可能です．

サブルーチン・関数・手続の使い方

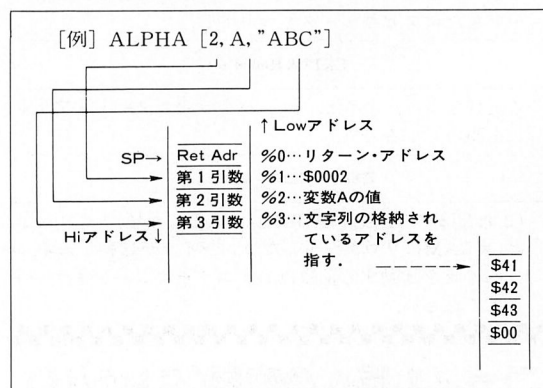
拡張K言語では，サブルーチン・関数・手続の区別がなく，**RETURN**文で終了しているルーチンは，そのいずれとしても使用が可能です．もし，サブルーチンとして使っているのであれば，呼び出しは**GOSUB**文で行ないます．これを関数・手続として使った場合，呼び出しは単にラベル名に[ ]で囲んで引数列を書きます．関数として使用するとき呼び出されたルーチンで，最後に実行された代入の値をラベル名に持ち帰ります．たとえば，階乗を値とする関数FACTは次のような記述が可能です．

```
10 'FACT;
20 ' IF %1=1 THEN A=1
30 '           ELSE A=%1*FACT[%1-1]
40 'FI
50 'RETURN
```

上の例で関数FACTはAに代入された値をFACTの値として持ち帰るので，引数を1としてFACTを呼ぶと，FACTの値が1となります．また，この例ではFACTの中で再びFACTを呼び出す，リカーシブコールも行なっています．ただし，このような処理をさせる場合，スタックがオーバーフローする危険があるので，充分な注意が必要です．

サブルーチン・関数・手続をコールした場合の引数は，スタックにその値がセットされる値渡しで，サブプログラム側に渡されます．ただし，文字列はその格納されているアドレスが，スタックにセットされるので，サブプログラム側ではそのアドレスから**PEEK**関数などで，文字列を読み出すことになります．なお，文字列の終わりは１バイトの０がセットされており，それ以後のプログラムと区別することができます．

サブプログラム側で引数を参照する場合は％変数を使いますが，これは前記のように，スタックの値を参照するため，コール側の引数を変化させることはできません．サブ側でコール側の引数を変化させたい場合，**ADR**関数でその変数のアドレスを渡し，**PEEK, POKE**などでその内容を変更することになります．

サブルーチンがコールされた時点でのスタックと，引数，％変数の対応は次のようになります．

外部ルーチンとのリンク

ユーザーがアセンブラなどで記述した手続や，システムのユーティリティ・ルーチンをコールするには，**DEF**文でその開始アドレスを定義し，呼び出しは一般のサブルーチンや関数と同様に行ないます．引数も同様にシステム・スタックに値がセットされて（サブルーチン・関数の使い方参照）渡されます．

ユーザールーチンからの値は，AccDにセットすれば，リターン後その値が外部関数の値となります．このときS，Uスタック，DPRは変更してはいけません．

Kコンパイラ・キーワード

キーワードはすべて予約語で，その意味はKコンパイラの仕様で定めています．キーワードの区切りは，空白または記号で，変数名をキーワードで始めた場合の動作は保障しません．

ABS, ADR, AND	INPUT
CALL, CHR$, CODE	NEXT
CONST, CPOKE	NOT
DEF	OR
ELSE, END	PEEK, POKE, PRINT
FI, FOR	REPEAT, RETURN, RND
LOW	SEX, SGN, SPC$
HEX$, **HEX2$**, HIGH	UNTILL
IF	WHILE, WEND
GET, GOSUB, GOTO	

エラーメッセージ

拡張Kコンパイラはコンパイル時の構文チェックを完全に行なわないので，プログラミングの際は次の点に充分注意してください．

(1)変数の誤りや，予約語との重複に注意する．
(2)サブルーチン・関数などの呼び出しでのパラメータの過不足．
(3)**CODE**文，**CALL**文の数値やアドレスの誤り．

コンパイル時のエラーメッセージとその意味を表2に示します．

表2　エラーメッセージ

DOUBLE DEF ERROR	ラベルの二重定義がある．
ELSE WITHOUT IF	ELSEに対応するIFがない．
FI WITHOUT IF	FIに対応するIFがない．
ILLEGAL ARGUMENT	手続，関数のパラメータに誤りがある．
ILLEGAL EXPRESSION	許されない式である．
ILLEGAL NESTING	REPEAT, WHILE, FOR, IF句の対応がおかしい．
NEXT WITHOUT FOR	NEXTに対応するFORがない．
SYNTAX ERROP	構文に誤りがある．
TOO MANY VALIABLES	変数の数が多すぎる．
UNDEFINED LABELS	未定義のラベルである．
UNTILL WITHOUT REPEAT	UNTILに対応するREPEATがない．
WEND WITHOUT WHILE	WENDに対応するWHILEがない．

最後に

使用例など豊富にだしたかったのですが，紙面に限りがあり，できませんでした．実際の使用例が，いろいろ載っているので，他のKコンパイラの記事を参考にしてください．

リファレンス

定　数

定　数	説　明
1文字定数	1文字定数は，ASCII文字1文字を『'』(シングル・クォート）で囲ったもので，その文字のASCIIコードを値とします． 例）IF GET='E' THEN PRINT "オワリ"：FI
文字定数	文字定数はASCII文字列を『"』(ダブル・クート）で囲ったもので，PRINT文中で出力する文字列として使ったり，関数や手続（サブルーチン）の引数に使ったりします． 例）"FM-7, FM-11"
10進定数	拡張K言語は整数型で，数値は−32768〜32767の範囲の整数が使えます．実数は使えません． 例）0 −32768
16進定数	16進定数は，10進定数と区別するために，先頭に『$』を付けます．桁数は4桁以下で，$0〜$FFFFの範囲です． 例）$0A $7FFF

ラベル

ラベル	説　明
ロケーションを指定するラベル	ラベル名は英字で始まる英数字列で，8文字までを認識します．GOTO文，GOSUB文で参照するラベルは，プログラム中の参照したい位置にラベルを書き，その後に『;』を書けば定義できます．ユーザーのマシン語サブルーチンなどの外部ルーチンをコールする場合は，DEF文でラベル定義をします． 例）SUB； FIRST；
数値を指定するラベル	ラベル名は英字で始まる英数字列で，8文字までを識別し，CONST文で数値を定義します．この数値は変更できません． 例）CONST DATA＝$3800

変　数	説　明
	変数はプログラム中で使う，変化する値を表す名前です．初期値は不定なので，使う前に必ず初期値をセットしてください．
単純変数	単純変数は英字で始まる英数字列で，8文字まで識別します．変数名を拡張K言語の予約語（たとえばFOR, NEXT, GOSUBなど）で始めた場合の動作は，保障しません．代入可能な値は，10進および16進定数の範囲で，−32768〜32767（$0〜$FFFF）です． 例）A FM-7 注）WENDA（WENDが含まれている）のように予約語で変数名を始めた場合の動作は，保障しない．また，ABCDEFGH5とABCDEFGH6は8文字以上なので，同一の変数とみなす．
配列変数	配列には2バイト配列と1バイト配列の2種類があります．配列の定義は，BASICのようにDIM文で宣言する必要はありませんが，配列を格納するメモリ・エリアを，プログラム中で指定する必要があります．また，配列は1次元のみなので，2次元以上の配列は，ユーザー側で要素番号を計算する必要があり，要素数の上限もユーザー管理です．使い方は『注意・その他』を参照してください．
%変数	これはサブルーチン側で，引数の参照をする場合に使います．参照は『%』と数値で行ない，数値は引数の順番を示します．たとえば，関数・手続側で，%3はその関数・手続の引数の3番目を参照します．このとき，%変数に数値を代入しても，メイン（呼び出し先）の引数列の変数を，変化させることはできません（値渡し）．関数・手続き側の値をメインに返すには，変化させたい変数のアドレスを引数で与え，そのアドレスの内容を変化させる必要があります． 例）%1 %5
算術演算子	＋，−，＊，／

式

式	説　　　　明
関係演算子 論理演算子	>, >=, <, <=, =, <> AND, OR

演算子の意味はBASICに準じ，優先順位は乗除，加減，論理の順です．ただし，１つの式に２個以上の論理演算子を書くことはできません．
除算の結果が整数でない場合，小数点以下は切り捨てられます．また，加減乗除算で，結果がオーバーフロー（桁あふれ，32768以上，または−32769以下）になった場合の結果は保障されません．

一般ステートメント

ステートメント	説　　　　明
コメント	コメントは『(＊』で始まり『＊)』で終わります．始まりと終わりの間には，キーボードから入力できるすべての文字を書くことができ，その文字列はプログラムには何の影響も与えません． コメントをプログラム中に注釈として入れることで，リストを見やすくすることができます． 例）(＊ 1モジ ニュウリョク ルーチン ＊)
代入	書式）変数名＝式 代入は，右辺の算術式，論理式，関数などの結果を左辺の変数に代入するものです． 例）A=5 KEKKA=$0F AND $D1 ADDRESS=ADR(X)
CONST	書式）CONST ラベル＝定数｛, ラベル＝定数｝… 数値を表わすラベルを宣語します『,』で区切ることで複数の定義を同時にできます． 例）CONST A=5 CONST XMAX=39, YMAX=24
DEF	書式）DEF ラベル＝定数｛, ラベル＝定数｝… 外部サブルーチンのスタート・アドレスを宣言します．定数は，ユーザー作成のマシン語ルーチンやシステムの持つサブルーチンの実行開始アドレスべす． 例）DEF TPREAD=$6E00 DEF TPREAD=$6E00, TPWRITE=$6F00
END	書式）END プログラムの実行を終了します．
PRINT	書式）PRINT 出力要素｛,出力要素｝… PRINT文は，それに続く出力要素の列をCRTに出力します． 出力要素とは，変数，式，文字列定数，関数，フォーマット関数で，『,』で区切って書くことができます．出力フォーマットはすべて，ユーザーが指定する必要があり，『／』で改行を，『#』で出力桁数を指定します．桁数の指定は， #〈式１〉,〈式２〉 で指定し，式２の値を式１の桁数とりで右詰めで出力します． 例）10' A=5；B=10 20' PRINT "A=", A,/ 30' PRINT "B=", #3, B,/ …このプログラムの出力は次のようになります． A=5 B=　10
IF-THEN～ELSE	書式）IF 条件(式)　THEN 処理１ ｛ELSE 処理２｝ F１条件が真（０以外）の場合処理１を，偽（０）の場合処理２を行ないます．FIはIFの終了を示し，１つのIF文に対しTHENとFIが必要です．ただし，ELSEと処理２は省略することができます． 例）変数Xが奇数か偶数かを判定するプログラム． 10' IF (X AND $01)=0 THEN PRINT"グウスウ" 20'　　　　　ELSE PRINT"キスウ" 30' FI 注意）IFとTHENは同じ行でなくてはいけません．
FOR-NEXT	書式）FOR 制御変数＝初期値　TO　終了値 ｛STEP 増分値｝；処理 NEXT FOR-NEXT間の処理を繰り返します．制御変数は最初に初期値が代入され，NEXTまでの処理を１回行うごとに，増分値が加算されます．この繰り返しを制御変数が終了値より大きくなるまで（増分値が負の場合小さくなるまで）行いますが，BASICと違い，最初から初期値が終了値より大きい（増分値が負の場合小さい）場合は処理を１度も行ないません． 初期値，終了値は定数または式で指定できますが，増分値は定数以外使えません．なお，STEPが省略された場合，1とみなされます．また，１つのFORに対して必ず対応するNEXTが必要で，FOR-NEXT間の処理で制御変数の値を変更することはできません． 例）1からNまでの数を表示します． FOR I=1 TO N PRINT I, / NEXT I 注１）制御変数の値が整数の範囲を超えるもうな指定をすると，無限ループとなる場合があります． 注２）FOR-NEXTから外へ，GOTO文，RETURN文で飛び出すことはできません．また，この中に外から飛び込むこともできません．
WHILE-WEND	書式）WHILE 条件 処理 WEND 条件が真であれば，WENDまでの処理を実行します．WENDで実行の制御は対応するWHILEに戻り，再び条件の判定を行なうので，条件が真の間はWHILE-WEND間の処理を繰り返します．条件が偽の場合，WEND以降に制御が移るので，最初から条件が偽の場合WHILE-WELD間の処理は１度も実行されません． 拡張Kコンパイラは，WHILE-WENDループの制御をスタックを使っていないマシン・コードに展開するので，WHILEに対するWENDは１箇所に限定されますが，WHILEループからGOTO文で抜けたり，飛び込むことが可能です．なお，条件は式または文で記述します． 例）IがNになるまで表示をします． WHILE I>N PRINT I, / I=I+1 WEND

ステートメント	説明
REPEAT-UNTILL	書式）REPEAT 処理 UNTIL 条件 条件が真になるまで，REPEAT-UNTIL間の処理を繰り返します．プログラムの実行は，UNTILに続く条件が偽であると，対応するREPEAT文にジャンプします．もし，条件が真であれば，実行の制御は条件以後の文に移り，ループにはなりませんが，条件に対する判定は UNTIL 文で行なうため，REPEAT-UNTIL間の処理は最低1回は行なわれます． 1つのREPEAT文に対するUNTIL文は1箇所で，WHILEループ同様GOTO文による飛び込み，飛び出しが可能です．条件は式または文で記述します． 例）Yキーが押されるまでループします． REPEAT A=GET UNTIL A='Y'
GOTO	書式）GOTO ラベル ラベルの定義されているロケーションに，実行の制御を移します．ラベルの定義はGOTO文の前後どちらにあってもかまいません． 例）GOTO DISPI
RETURN	書式）RETURN サブルーチン，関数，手続を終わり，現在実行しているルーチンを呼び出した文の直後に復帰します．
POKE	書式）POKE 式1，式2 式1で指定するアドレスに，式2の値の下位1バイトを書き込みます． 例）POKE $4000，$FF ………$400番地に$FFを書き込みます．
GOSUB	書式）GOSUB ラベル ラベルの定義されているサブルーチンに，実行の制御を移します．サブルーチンはRETURN文で終わっていなければなりません． 例）GOSUB LABEL
関数・手続の呼び出し	書式）ラベル［{引数1，…，引数n}］ ラベルの定義されている関数・手続に，引数の値を持って実行の制御を移します．関数・手続はRETURN文で終わっていなければなりません． 関数・手続を呼び出すには，ラベル名の後に"［ ］"で囲んで引数列を書きます．引数がないときも，"［ ］"を付けてください．引数は，","で区切って複数個書くことができます．引数には，定数，変数，式が使えます．詳しくは『サブルーチン・関数・手続の使い方』を参照してください．
CPOKE	書式）CPOKE 変数，式1{#}，式2{#}，……，式n{#} 変数で指定するアドレスから，式の列のデータを書き込みます．変数の値は書き込んだデータ数だけ増加します．式の値は下位1バイトを書き込みますが，式の後に『#』がある場合は2バイトのデータを書き込みます．式が2個以上ある場合，式の間は『，』で区切ってください． 例）CPOKE ADR，$00，D1#，'B'……変数ADRの内容で指定されるアドレスから順に，$00，変数D1の値2バイト，『B』のASCIIコード$42を書き込みます．
CODE	書式）CODE 数値1{#}，{数値2{#}，……，数値n{#}} 数値1～数値nの値をプログラム中に展開します．数値が2個以上ある場合は『，』で区切って記述してください．式の値は下位1バイトが展開されますが，式の後に『#』を書くと2バイトが展開されます． 例）CONST文との組み合わせで，疑似的なアセンブラ記述をした例． CONST LDA=$86，RTS=$39 CODE LDA，$0F，RTS
CALL	書式）CALL 定数 定数で指定したアドレスを，マシン語のコール命令でコールします．定数で指定したアドレスから始まるマシン語サブルーチンがRTS命令で終わっていれば，CALL文の次に制御が戻ります． 例）CALL $2000

関　数

関数	説明
拡張Kコンパイラには数値関数，入力関数，フォーマット関数の3種類の関数があります．	
ABS	ABSは数値関数で，式の値の絶対値を値とします． 例）D=ABS(−5) ……変数Dの値は5になります．
ADR	書式）ADR(単純変数) ADRは数値関数で，単純変数の値が格納されているアドレスを値とします． 例）POKE ADR(D)，10 変数Dの値は10になります．
HIGH	書式）HIGH(式) HIGHは数値関数で，式の値の上位1バイトを値とします． 例）H=HIGH($1234) ……変数Hの値は$12になります．
LOW	書式）LOW(式) LOWは数値関数で，式の値の下位バイトを値とします． 例）L=LOW($1234) ……変数Lの値は$34になります．
NOT	書式）NOT(式) NOTは数値関数で，式の値の論理否定を値とします．真は0，偽は1となります．
PEEK	書式）PEEK(式) PEEKは数値関数で，式で指定されるメモリ・アドレスの内容を値とします． 例）D=PEEK($6000) ……変数Dの値は$6000の値と同じになります．
RND	書式）RND(式) RNDは数値関数で，0～(式-1)の範囲内の乱数を値とします． 例）D=RND(5) ……変数Dに0～4の乱数を代入します．
SEX	書式）SEX(式) SEXは数値関数で，1バイトの数を2バイトの数に拡張します． 例）D=SEX($FF) ……変数Dは$FFFF(=−1)になります．
SGN	書式）SGN(式) SGNは数値関数で，式の符号により1，

関　　数	説　　　明
	0，－1の値をとります．式の値が1以上であれば関数の値は1，－1以下であればー1，0であれば0になります． 例）IF SGN(X)=－1 THEN PRINT" Xハ　マイナス"; F1 ……変数Xが負の場合，その値のメッセージを出力します．
INPUT	書式）INPUT INPUTは入力関数で，キーボードから入力された値数を値とします．入力できる数値は－32768～32767の範囲の整数か，$0000～$FFFFの16進数で，16進数の場合は先頭に『$』をつけて入力します． 例）ID=INPUT ……変数IDにキーボードから入力した値を代入します．
GET	書式）GET GETは入力関数で，キーボードをスキャンして押されていたキーのASCIIコードを値とします．キーが押されていない場合は，0が値となります． 例）KD=GET ……変数KDに押されていたキーのASCIIコードを代入します．

関　　数	説　　　明
CHR$	書式）CHR$(式) CHR$はフォーマット関数で，式の値をASCIIコードとみなし，その値に対応する文字を与えます． 例）PRINT CHR$($41) ……『A』を表示します．
HEX$	書式）HEX$(式) HEX$はフォーマット関数で，式の値を16進数4桁に変換して与えます． 例）PRINT HEX$(15) ……表示は『000F』となります．
HEX2$	書式）HEX2$(式) HEX2$はフォーマット関数で，式の値を16進数2桁に変換して与えます． 例）PRINT HEX2$(15) ……表示は『0F』となります．
SPC$	書式）SPC$(式) SPC$はフォーマット関数で，式の値の空白を与えます． 例）PRINT "AB", SPC$(3), "CD" ……表示は『AB␣␣␣CD』となります．

リスト5　ランタイム・ルーチン（KV3.6）

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4D00 16 02 35 16 02 79 17 02 99 20 0C 35 10 20 02 8D :B0
4D10 EF A6 80 26 FA 6E 84 86 0D 8D E5 86 0A 20 E1 36 :F3
4D20 06 A6 43 3D 34 06 EC 41 3D EB E4 E7 E4 A6 C4 E6 :BA
4D30 43 3D EB E0 A6 E0 1E 89 33 44 39 8E 00 00 34 10 :FA
4D40 34 10 34 10 4D 2A 04 8D 40 63 61 6C 60 2B 40 58 :23
4D50 49 2A F8 44 56 ED 64 EC C1 2A 04 8D 2C 63 61 6A :18
4D60 60 2B 1E 58 49 2A F8 44 56 A3 64 20 06 58 49 24 :F8
4D70 F8 E3 64 1C FE 2B 02 1A 01 69 63 69 62 6A 60 2A :2C
4D80 EC EC 62 6D 61 32 66 2A 05 43 53 C3 00 01 39 32 :94
4D90 66 CC 7F FF 39 4D 2B F1 39 A3 C1 2F 2C CC 00 00 :16
4DA0 39 A3 C1 2E 24 CC 00 00 39 A3 C1 2D 1C CC 00 00 :6D
4DB0 39 A3 C1 2C 14 CC 00 00 39 A3 C1 27 0C CC 00 00 :45
4DC0 39 A3 C1 26 04 CC 00 00 39 CC 00 01 39 83 00 00 :55
4DD0 27 F7 CC 00 00 39 17 FF 2A 1F 89 4F 39 8D 15 16 :4B
4DE0 FF A7 8E 04 3D 17 FF 1E A6 80 81 24 27 2B 81 2D :74
4DF0 27 EB 30 1F CC 00 00 34 06 A6 84 82 30 2B 18 81 :07
---------------------------------------------------------
Sum: 6D FD 3F 30 9F 6C AE 95 2D B2 5E EE 0F 01 0C BF :2D


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4E00 09 22 14 A7 84 EC 60 58 49 58 49 E3 60 58 49 EB :C7
4E10 80 89 00 ED 60 20 E2 35 86 CC 00 00 34 06 A6 84 :43
4E20 82 30 2B F3 81 09 23 06 82 07 81 0F 22 E9 A7 84 :D2
4E30 EC 60 58 49 58 49 58 49 58 49 EB 80 ED 60 20 DE :86
4E40 8E 00 00 36 10 36 10 36 10 36 10 4D 2A 05 63 40 :C5
4E50 17 00 D7 58 49 ED 46 8E 00 0F C6 05 1C FE A6 C5 :AF
4E60 49 81 09 22 04 1C FE 20 04 82 0A 1A 01 A7 C5 5A :A4
4E70 26 EC 68 47 69 46 24 02 6C 45 30 1F 26 DC C6 01 :5F
4E80 A6 C5 26 05 5C C1 05 26 F7 39 8D B4 34 04 EB 49 :BB
4E90 C2 06 23 02 8D 20 35 04 6D 40 2A 05 86 2D 17 FE :77
4EA0 5F A6 C5 8B 30 17 FE 58 5C C1 06 26 F4 33 4A 39 :E5
4EB0 33 5E 8D 8C 20 E2 86 20 17 FE 45 5A 26 FA 39 34 :93
4EC0 04 8D 04 35 04 1F 98 1F 89 44 44 44 44 8D 04 1F :ED
4ED0 98 84 0F 8B 30 81 39 23 02 8B 07 16 FE 22 1F 98 :44
4EE0 20 F9 33 5C A6 45 E6 47 3D ED 42 EC 45 3D EB 42 :C7
4EF0 89 00 ED 41 A6 44 E6 47 3D E3 41 ED 41 A6 44 E6 :2D
---------------------------------------------------------
Sum: 4A 81 AD 42 3C E6 90 34 05 57 95 69 AC 1D 21 C4 :A8


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4F00 46 3D EB 41 89 00 ED C4 39 34 06 EC 8C 1A 8E 3D :B9
4F10 09 36 16 8D CD EC 42 33 48 C3 00 03 ED 8C 09 35 :D5
4F20 10 36 16 8D BD 33 48 39 1C 2C 6A 49 16 FE 5A 20 :E3
4F30 7C CF 34 3B 8C 00 00 00 20 02 20 13 34 31 8D 26 :B3
4F40 10 8E FC 82 8E 03 01 AF A1 A7 A4 8D 2E 35 B1 34 :1E
4F50 05 F6 FD 02 54 25 FA B7 FD 01 C6 00 F7 FD 00 CA :A6
4F60 40 F7 FD 00 35 85 34 02 B6 FD 05 2B FB 1A 40 86 :E2
4F70 80 B7 FD 05 B6 FD 05 2A FB 35 82 7F FD 05 39 34 :BB
4F80 05 C6 12 8D 5D 8D DF CC 29 00 FD FC 82 8D EC 8D :A9
4F90 D5 8D 02 35 85 C6 80 B6 FC 83 FA FC 80 F7 FC 80 :82
4FA0 20 D9 34 35 C6 13 8D 3A 8D BC CC 04 02 FD FC 82 :98
4FB0 CC 11 07 FD FC 84 8D C3 8D AC F6 FC 83 26 2C F6 :A7
4FC0 FC 84 27 12 5A 5A 5A 27 0D 10 8E FC 88 A6 A0 27 :8A
4FD0 05 A7 80 5A 26 F7 C6 0D E7 84 8D B9 C6 12 8D 02 :8E
4FE0 35 B5 8D 82 86 0C FD FC 82 20 90 4F 1F 8B 8D A5 :E1
4FF0 B6 FD 01 81 FF 27 06 7D FD 0F 7E 8F D0 7E 94 8D :66
---------------------------------------------------------
Sum: 62 C4 C2 82 15 37 47 EE BE AD 63 0D A4 8E 06 50 :4E
```

リスト6　拡張Kコンパイラ（KV3.6）

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2000 34 7F C6 4C 1F 9B CE 4C 00 10 DF FE 12 12 12 12 :CE
2010 BD 4D 17 BD 4D 0B 4B 20 43 4F 4D 50 49 4C 45 52 :FC
2020 20 56 45 52 53 49 4F 4E 20 33 2E 36 00 BD 4D 17 :1E
2030 BD 4D 0B 43 4F 50 59 52 49 47 48 54 20 31 39 38 :90
2040 32 20 42 59 20 43 4F 4D 50 41 43 2E 00 BD 4D 17 :0F
2050 BD 4D 17 BD 4D 17 BD 4D 0B 43 4F 4D 50 49 4C 49 :64
2060 4E 47 2E 2E 2E 00 BD 4D 17 BD 4D 17 BD 4D 17 17 :99
2070 21 70 CC 42 00 DD 00 CC 00 01 DD 02 CC 50 00 DD :21
2080 04 CC 43 00 DD 06 CC 00 FF 9E 06 E7 84 CC 4B 8E :75
2090 DD 08 CC FF 00 9E 08 ED 84 CC 4B 00 DD 0A CC 00 :91
20A0 00 DD 0C CC 00 0D DD 0E CC 00 7F DD 10 CC 00 00 :B1
20B0 DD 12 CC 00 00 DD 14 CC 34 7F 9E 04 ED 81 CC C6 :CD
20C0 4C 12 12 ED 81 CC 1F 9B 12 12 ED 81 CC CE 4C 12 :EE
20D0 12 ED 81 CC 00 10 12 12 ED 81 CC DF FE ED 81 9F :A4
20E0 04 17 20 A2 9E 16 4F E6 84 93 10 10 26 01 9D 17 :D8
20F0 20 FC CC 10 DE 9E 04 ED 84 CC FE 35 9E 04 ED 02 :79
---------------------------------------------------------
Sum: 6C 68 E6 5A 83 94 D3 06 A8 F6 93 D9 40 D2 C7 25 :0C


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2100 CC FF 00 9E 04 ED 04 CC 00 00 9E 04 E7 06 DC 04 :99
2110 C3 00 05 DD 04 9E 08 EC 84 83 FF 00 10 27 00 94 :0C
2120 17 20 CC BD 4D 17 BD 4D 0B 49 4C 4C 45 47 41 4C :33
2130 20 4E 45 53 54 49 4E 47 2E 00 BD 4D 17 BD 4D 0B :9C
2140 4C 49 4E 45 00 CC 00 06 36 06 9E 08 EC 08 BD 4E :DB
2150 8A 9E 08 EC 84 83 FF FF 10 26 00 08 BD 4D 0B 20 :94
2160 46 4F 52 00 9E 08 EC 84 83 FF FE 10 26 00 07 BD :77
2170 4D 0B 20 49 46 00 9E 08 EC 84 83 FF FD 10 26 00 :D2
2180 0B BD 4D 0B 20 52 45 50 45 41 54 00 9E 08 EC 84 :17
2190 83 FF FC 10 26 00 0A BD 4D 0B 20 57 48 49 4C 45 :6C
21A0 00 BD 4D 17 DC 08 C3 00 0A DD 08 9E 08 EC 84 83 :50
21B0 FF 00 26 89 DC 06 DD 18 CC 00 00 DD 1A 9E 18 4F :4D
21C0 E6 84 83 00 FF 10 27 00 6D 9E 18 EC 84 83 00 01 :3A
21D0 10 26 00 58 DC 1A 83 00 00 10 26 00 1F 17 20 0F :A2
21E0 BD 4D 17 BD 4D 0B 55 4E 44 45 46 49 4E 45 44 20 :E8
21F0 4C 41 42 45 4C 53 20 3A 00 BD 4D 17 BD 4D 0B 20 :63
---------------------------------------------------------
Sum: BB 5F 76 1A 83 2A AE 8A 8B 54 12 DA D5 9D A2 05 :73
```

リスト6　拡張Kコンパイラ(KV3.6)

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
2200	00	CC	00	04	DD	1C	CC	00	0B	36	06	DC	1C	16	00	11	:FB
2210	DC	1C	9E	18	E6	8B	4F	BD	4E	DE	DC	1C	C3	00	01	DD	:F0
2220	1C	A3	C4	2F	EB	33	42	CC	00	01	DD	1A	DC	18	C3	00	:8D
2230	0C	DD	18	16	FF	87	BD	4D	17	BD	4D	17	BD	4D	0B	4F	:43
2240	42	4A	45	43	54	20	50	52	4F	47	52	41	4D	20	53	49	:5C
2250	5A	45	20	3A	00	DC	04	A3	8D	FE	22	DD	1E	12	BD	4E	:41
2260	B0	BD	4D	0B	20	28	24	00	DC	1E	BD	4E	BF	BD	4D	0B	:0A
2270	29	00	BD	4D	17	BD	4D	17	BD	4D	0B	43	4F	4D	50	49	:F8
2280	4C	45	20	45	4E	44	2E	00	17	1F	65	39	17	12	33	9E	:84
2290	16	4F	E6	84	36	06	CC	00	3B	BD	4D	B9	36	06	9E	16	:C5
22A0	4F	E6	84	36	06	CC	00	3B	BD	4D	B9	AA	C0	EA	C0	10	:E3
22B0	27	00	0A	DC	16	C3	00	01	DD	16	16	FF	CF	9E	16	4F	:C1
22C0	E6	84	93	0E	10	26	00	03	16	FE	16	DC	14	36	06	CC	:66
22D0	00	01	BD	4D	B9	36	06	30	8C	02	20	03	28	2A	00	34	:67
22E0	10	17	10	EE	32	62	AA	C0	EA	C0	10	27	00	5D	CC	00	:2D
22F0	01	DD	14	DC	16	C3	00	01	DD	16	9E	16	4F	E6	1F	36	:D9
Sum:	48	A7	F1	36	E9	9C	89	12	3A	97	AD	8F	58	FA	14	71	:1A

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
2300	06	CC	00	2A	BD	4D	B9	36	06	9E	16	4F	E6	84	36	06	:A4
2310	CC	00	29	BD	4D	B9	A4	C0	E4	C0	36	06	9E	16	4F	E6	:E5
2320	1F	36	06	DC	0E	BD	4D	B9	AA	C0	EA	C0	27	C5	9E	16	:BC
2330	4F	E6	1F	93	0E	10	26	00	03	16	FD	A5	DC	16	C3	00	:9B
2340	01	DD	16	CC	00	00	DD	14	16	FF	41	30	8C	02	20	06	:EB
2350	50	52	49	4E	54	00	34	10	17	10	77	32	62	36	06	30	:6F
2360	8C	02	20	02	3F	00	34	10	17	10	67	32	62	AA	C0	EA	:A9
2370	C0	10	27	01	BF	17	11	4A	9E	16	4F	E6	84	83	00	2C	:45
2380	10	26	00	0A	DC	16	C3	00	01	DD	16	16	FF	E7	9E	16	:99
2390	4F	E6	84	83	00	2F	10	26	00	14	CC	4D	17	34	06	17	:36
23A0	10	84	32	62	DC	16	C3	00	01	DD	16	16	FF	C7	9E	16	:61
23B0	4F	E6	84	83	00	23	10	26	00	3E	DC	16	C3	00	01	DD	:66
23C0	16	17	11	75	9E	16	4F	E6	84	83	00	2C	10	27	00	03	:09
23D0	16	1C	5B	CC	36	06	9E	04	ED	84	DC	04	C3	00	02	DD	:2A
23E0	04	DC	16	C3	00	01	DD	16	17	11	4E	CC	4E	8A	34	06	:01
23F0	17	10	33	32	62	16	FF	7D	9E	16	4F	E6	84	83	00	22	:92
Sum:	E2	BE	E3	1B	66	9B	95	F6	A1	A3	EE	A5	D8	F0	45	76	:84

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
2400	10	26	00	61	CC	4D	0B	34	06	17	10	1A	32	62	DC	16	:BC
2410	C3	00	01	DD	16	9E	16	4F	E6	84	9E	04	E7	84	DC	04	:11
2420	C3	00	01	DD	04	9E	16	4F	E6	84	36	06	CC	00	22	BD	:F9
2430	4D	B9	36	06	9E	16	4F	E6	84	36	06	DC	0E	BD	4D	B9	:98
2440	AA	C0	EA	C0	27	C8	CC	00	00	9E	04	E7	1F	9E	16	4F	:7A
2450	E6	84	93	0E	10	26	00	03	16	FC	86	DC	16	C3	00	01	:92
2460	DD	16	16	FF	10	30	8C	02	20	06	43	48	52	24	28	00	:25
2470	34	10	17	0F	5D	32	62	10	27	00	10	17	10	9F	CC	4E	:82
2480	DE	34	06	17	0F	A0	32	62	16	FE	EA	30	8C	02	20	06	:54
2490	48	45	58	24	28	00	34	10	17	0F	37	32	62	10	27	00	:9D
24A0	10	17	10	79	CC	4E	BF	34	06	17	0F	7A	32	62	16	FE	:0B
24B0	C4	30	8C	02	20	07	48	45	58	32	24	28	00	34	10	17	:67
24C0	0F	10	32	62	10	27	00	10	17	10	52	CC	4E	C5	34	06	:8C
24D0	17	0F	53	32	62	16	FE	9D	30	8C	02	20	06	53	50	43	:88
24E0	24	28	00	34	10	17	0E	EA	32	62	10	27	00	10	17	10	:A1
24F0	2C	CC	4E	B6	34	06	17	0F	2D	32	62	16	FE	77	9E	16	:5C
Sum:	F4	1C	AF	31	01	3E	D0	5E	E4	7B	E1	4F	FC	0E	D7	B8	:85

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
2500	4F	E6	84	36	06	CC	00	3B	BD	4D	B9	36	06	9E	16	4F	:FE
2510	E6	84	36	06	DC	0E	BD	4D	B9	AA	C0	EA	C0	10	27	00	:9E
2520	03	16	FD	68	17	10	12	CC	4E	B0	34	06	17	0E	F7	32	:09
2530	62	16	FE	41	30	8C	02	20	06	43	4F	4E	53	54	00	34	:56
2540	10	17	0E	8E	32	62	10	27	00	08	CC	03	00	DD	20	16	:78
2550	00	19	30	8C	02	20	04	44	45	46	00	34	10	17	0E	72	:A5
2560	32	62	10	27	00	9B	CC	02	00	DD	20	DC	16	83	00	01	:A7
2570	DD	16	DC	16	C3	00	01	DD	16	17	0D	17	DC	22	83	00	:58
2580	03	10	27	00	1B	BD	4D	17	BD	4D	0B	44	4F	55	42	4C	:01
2590	45	20	44	45	46	20	45	52	52	4F	52	2E	00	16	1A	A8	:E4
25A0	9E	16	4F	E6	84	83	00	3D	10	27	00	03	16	0C	CD	DC	:32
25B0	08	83	00	02	DD	08	DC	18	9E	08	ED	84	DC	16	C3	00	:32
25C0	01	DD	16	17	0F	73	9E	08	EC	84	DD	18	DC	08	C3	00	:3F
25D0	02	DD	08	DC	1A	83	00	01	10	27	00	03	16	0C	9D	DC	:36
25E0	20	9E	18	ED	84	DC	24	9E	18	ED	02	DC	04	83	00	03	:52
25F0	DD	04	9E	16	4F	E6	84	83	00	2C	10	27	FF	74	16	FC	:B9
Sum:	A7	63	6D	5F	DE	B3	66	A6	F6	BB	2E	B5	68	41	47	E9	:E0

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
2600	8B	30	8C	02	20	06	47	4F	54	4F	20	00	34	10	17	0D	:30
2610	C1	32	62	10	27	00	17	17	0C	79	CC	00	16	9E	04	E7	:AA
2620	84	DC	04	C3	00	03	DD	04	17	0E	AE	16	FC	5E	30	8C	:0A
2630	02	20	07	47	4F	53	55	42	20	00	34	10	17	0D	93	32	:F6
2640	62	10	27	02	00	17	0C	4B	17	00	03	16	FC	3E	DC	08	:57
2650	83	00	04	DD	08	DC	18	9E	08	ED	84	DC	22	9E	08	ED	:08
2660	02	CC	00	00	DD	26	9E	16	4F	E6	84	83	00	5B	10	26	:52
2670	01	48	9E	16	4F	E6	01	83	00	5D	10	26	00	0A	DC	16	:45
2680	C3	00	02	DD	16	16	01	32	DC	16	C3	00	01	DD	16	DC	:86
2690	26	C3	00	01	DD	26	9E	16	4F	E6	84	83	00	22	10	26	:35
26A0	00	A5	CC	30	8C	9E	04	ED	84	CC	02	20	9E	04	ED	02	:BF
26B0	DC	04	C3	00	05	DD	04	DC	04	DD	28	CC	00	00	DD	2A	:41
26C0	DC	16	C3	00	01	DD	16	DC	2A	C3	00	01	DD	2A	9E	16	:2E
26D0	4F	E6	84	9E	04	E7	84	DC	04	C3	00	01	DD	04	9E	16	:FF
26E0	4F	E6	84	36	06	CC	00	22	BD	4D	B9	36	06	9E	16	4F	:E5
26F0	E6	84	36	06	DC	0E	BD	4D	B9	AA	C0	EA	C0	27	C1	DC	:2B
Sum:	DF	54	54	F9	35	B0	51	66	5C	28	D3	52	9A	50	B1	68	:C8

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
2700	2A	36	06	CC	00	7F	BD	4D	A9	36	06	9E	16	4F	E6	84	:0D
2710	36	06	DC	0E	BD	4D	B9	AA	C0	EA	C0	10	27	00	03	16	:4D
2720	19	7E	CC	00	00	9E	04	E7	1F	DC	2A	9E	28	E7	1F	CC	:A9
2730	36	10	9E	04	ED	84	DC	16	C3	00	01	DD	16	DC	04	C3	:A5
2740	00	02	DD	04	16	00	11	17	0D	EF	CC	36	06	9E	04	ED	:B4
2750	84	DC	04	C3	00	02	DD	04	9E	16	4F	E6	84	83	00	2C	:26
2760	10	27	FF	24	9E	16	4F	E6	84	83	00	5D	10	27	00	03	:E1
2770	16	19	2D	DC	16	C3	00	01	DD	16	DC	26	83	00	00	10	:9A
2780	2F	00	37	CC	00	34	9E	04	E7	1E	CC	00	02	DD	1C	DC	:B0
2790	26	36	06	DC	1C	16	00	1C	CC	37	10	9E	04	ED	84	CC	:7E
27A0	34	10	9E	04	ED	02	DC	04	C3	00	04	DD	04	DC	1C	C3	:18
27B0	00	01	DD	1C	A3	C4	2F	E0	33	42	CC	00	17	9E	04	E7	:51
27C0	84	DC	04	C3	00	03	DD	04	9E	08	EC	84	DD	18	9E	08	:BC
27D0	EC	02	DD	22	DC	08	C3	00	04	DD	08	9E	18	4F	E6	84	:EC
27E0	83	00	02	10	27	00	06	17	0C	EF	16	00	0F	CC	00	BD	:82
27F0	9E	04	E7	1D	9E	18	EC	02	9E	04	ED	1E	DC	26	83	00	:7C
Sum:	73	11	DB	7F	C1	FC	CE	17	4C	09	8B	83	99	F7	D7	F0	:3A

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
2800	00	10	2F	00	3F	CC	00	32	9E	04	E7	84	DC	04	C3	00	:2C
2810	02	DD	04	DC	26	83	00	08	10	2C	00	12	CC	00	60	36	:20
2820	06	DC	26	58	49	E3	C1	9E	04	E7	1F	16	00	16	CC	00	:ED
2830	E9	9E	04	E7	1F	DC	26	58	49	9E	04	ED	84	DC	04	C3	:EA
2840	00	02	DD	04	39	30	8C	02	20	07	52	45	54	55	52	4E	:E1
2850	00	34	10	17	0B	7C	32	62	10	27	00	11	CC	00	39	9E	:61
2860	04	E7	84	DC	04	C3	00	01	DD	04	16	FA	1F	30	8C	02	:E1
2870	20	07	52	45	50	45	41	54	00	34	10	17	0B	54	32	62	:36
2880	10	27	00	1D	DC	08	83	00	0A	DD	08	DC	12	9E	08	ED	:2B
2890	08	CC	FF	FD	9E	08	ED	84	DC	04	9E	08	ED	02	16	F9	:6B
28A0	EB	30	8C	02	20	07	55	4E	54	49	4C	20	00	34	10	17	:D7
28B0	0B	20	32	62	10	27	00	78	9E	08	EC	84	83	FF	FD	10	:13
28C0	27	00	1F	BD	4D	17	BD	4D	0B	55	4E	54	49	4C	20	57	:7F
28D0	49	54	48	4F	55	54	20	52	45	50	45	41	54	2E	00	16	:02
28E0	F8	33	17	0C	54	17	17	F0	9E	08	EC	02	93	04	83	FF	:6D
28F0	81	10	2F	00	1A	DC	2C	9E	04	E7	84	DC	04	C3	00	02	:94
Sum:	0C	65	8A	ED	1F	5E	CB	60	D2	E1	63	FB	2C	E3	0A	C4	:7E

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
2900	DD	04	9E	08	EC	02	93	04	9E	04	E7	1F	16	00	17	DC	:BD
2910	2C	9E	04	ED	84	DC	04	C3	00	04	DD	04	9E	08	EC	02	:5B
2920	93	04	9E	04	ED	1E	DC	08	C3	00	0A	DD	08	16	F9	5C	:45
2930	30	8C	02	20	07	57	48	49	4C	45	20	00	34	10	17	0A	:E3
2940	91	32	62	10	27	00	36	DC	08	83	00	0A	DD	08	DC	12	:D6
2950	9E	08	ED	08	CC	FF	FC	9E	08	ED	84	DC	04	9E	08	ED	:EC
2960	02	17	0B	D5	17	17	71	DC	2C	9E	04	ED	84	DC	04	C3	:56
2970	00	04	DD	04	DC	04	9E	08	ED	04	16	F9	0F	30	8C	02	:38
2980	20	05	57	45	4E	44	00	34	10	17	0A	46	32	62	10	27	:C9
2990	00	58	9E	08	EC	84	83	FF	FC	10	27	00	1D	BD	4D	17	:61
29A0	BD	4D	0B	57	45	4E	44	20	57	49	54	48	4F	55	54	20	:B7
29B0	57	48	49	4C	45	2E	00	16	F7	5B	CC	00	16	9E	04	E7	:7A
29C0	84	DC	04	C3	00	03	DD	04	9E	08	EC	02	93	04	9E	04	:D8
29D0	ED	1E	9E	08	EC	04	DD	2E	DC	04	93	2E	9E	2E	ED	1E	:24
29E0	DC	08	C3	00	0A	DD	08	16	F8	A2	30	8C	02	20	04	49	:71
29F0	46	20	00	34	10	17	09	DA	32	62	10	27	00	48	17	0B	:D9
Sum:	C4	9B	27	F9	14	AC	8E	01	D4	3A	9C	3D	4B	8C	E2	C3	:31

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
2A00	38	17	16	D4	30	8C	02	20	05	54	48	45	4E	00	34	10	:8F
2A10	17	09	BF	32	62	DC	08	83	00	0A	DD	08	DC	12	9E	08	:5D
2A20	ED	08	CC	FF	FE	9E	08	ED	84	DC	2C	9E	04	ED	84	DC	:CC
2A30	04	C3	00	04	DD	04	DC	04	9E	08	ED	02	CC	00	00	9E	:8B
2A40	08	ED	04	16	F8	46	30	8C	02	20	05	45	4C	53	45	00	:59
2A50	34	10	17	09	7D	32	62	10	27	00	4A	9E	08	EC	84	83	:8F
2A60	FF	FE	10	27	00	1A	BD	4D	17	BD	4D	0B	45	4C	53	45	:AD
2A70	20	57	49	54	48	4F	55	54	20	49	46	2E	00	16	F6	95	:D2
2A80	9E	08	EC	02	DD	2E	CC	00	16	9E	04	E7	84	DC	04	C3	:31
2A90	00	03	DD	04	DC	04	9E	08	ED	02	DC	04	93	2E	9E	2E	:C6
2AA0	ED	1E	16	F7	E7	30	8C	02	20	03	46	49	00	34	10	17	:CA
2AB0	09	20	32	62	10	27	00	3B	9E	08	EC	84	83	FF	FE	10	:D5
2AC0	27	00	18	BD	4D	17	BD	4D	0B	46	49	20	57	49	54	48	:60
2AD0	4F	55	54	20	49	46	2E	00	16	F6	3A	9E	08	EC	02	DD	:8C
2AE0	2E	DC	04	93	2E	9E	2E	ED	1E	DC	08	C3	00	0A	DD	08	:3C
2AF0	16	F7	99	30	8C	02	20	05	46	4F	52	20	00	34	10	17	:EB
Sum:	E9	AE	2F	A2	2A	71	C1	55	CD	7A	0F	62	8C	50	5B	4B	:53

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
2B00	08	D0	32	62	10	27	00	F4	17	07	88	9E	16	4F	E6	84	:AA
2B10	83	00	3D	10	27	00	03	16	07	62	DC	22	83	00	03	10	:0D
2B20	26	00	20	CC	01	00	9E	18	ED	84	DC	0C	9E	18	ED	02	:C7
2B30	DC	0C	C3	00	02	DD	0C	DC	0C	83	01	02	10	26	00	03	:3D
2B40	16	15	78	9E	18	EC	02	DD	30	DC	16	C3	00	01	DD	16	:FD
2B50	17	09	E6	30	8C	02	20	03	54	4F	00	34	10	17	08	72	:5F
2B60	32	62	83	00	00	10	26	00	03	16	07	10	CC	00	DD	9E	:C4
2B70	04	E7	84	DC	30	9E	04	E7	01	DC	04	C3	00	02	DD	04	:8B
2B80	DC	08	83	00	0A	DD	08	DC	12	9E	08	ED	08	CC	FF	FF	:A9
2B90	9E	08	ED	84	DC	30	9E	08	ED	02	17	09	9C	CC	36	06	:7C
2BA0	9E	04	ED	84	CC	00	DC	9E	04	E7	02	DC	30	9E	04	E7	:DB
2BB0	03	CC	00	16	9E	04	E7	04	DC	04	C3	00	07	DD	04	DC	:D9
2BC0	04	9E	08	ED	04	30	8C	02	20	05	53	54	45	50	00	34	:EE
2BD0	10	17	07	FE	32	62	10	27	00	18	17	09	5C	DC	1A	83	:04
2BE0	00	01	10	27	00	03	16	06	93	DC	24	9E	08	ED	06	16	:99
2BF0	00	07	CC	00	01	9E	08	ED	06	16	F6	90	30	8C	02	20	:E7
Sum:	1F	E0	FF	18	95	E4	1C	67	37	27	CA	F5	D7	5F	D4	78	:B1

リスト6　拡張Kコンパイラ(KV3.6)

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2C00 05 4E 45 58 54 00 34 10 17 07 C7 32 62 10 27 00 :38
2C10 ED 9E 08 EC 84 83 FF FF 10 27 00 1B BD 4D 17 BD :B4
2C20 4D 0B 4E 45 58 54 20 57 49 54 48 4F 55 54 20 46 :51
2C30 4F 52 2E 00 16 F4 DE CC 00 DC 9E 04 E7 84 9E 08 :12
2C40 EC 02 9E 04 E7 01 CC 00 C3 9E 04 E7 02 CC 00 DD :3B
2C50 9E 04 E7 05 9E 08 EC 02 9E 04 E7 06 CC 00 A3 9E :BE
2C60 04 E7 07 CC 00 C4 9E 04 E7 08 DC 04 C3 00 09 DD :9C
2C70 04 9E 08 EC 06 9E 04 ED 1A 9E 08 EC 06 83 00 00 :60
2C80 10 2F 00 08 CC 10 2F DD 32 16 00 05 CC 10 2C DD :61
2C90 32 9E 08 EC 04 DD 2E DC 04 83 00 02 93 2E 9E 2E :C5
2CA0 ED 1E 9E 08 EC 04 93 04 83 FF 81 10 2F 00 24 DC :7A
2CB0 32 9E 04 E7 84 9E 08 EC 04 93 04 83 00 02 9E 04 :93
2CC0 E7 01 CC 33 42 9E 04 ED 02 DC 04 C3 00 04 DD 04 :42
2CD0 16 00 21 DC 32 9E 04 ED 84 9E 08 EC 04 93 04 83 :08
2CE0 00 04 9E 04 ED 02 CC 33 42 9E 04 ED 04 DC 04 C3 :0C
2CF0 00 06 DD 04 DC 08 C3 00 0A DD 08 16 F5 8E 30 8C :D2
-------------------------------------------------------
Sum: 7E 68 6F 44 4E 0B 1A DB 61 C6 19 C9 7D C5 49 24 :9F
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2D00 02 20 05 50 4F 4B 45 00 34 10 17 06 C5 32 62 10 :20
2D10 27 00 BB 17 08 23 9E 16 4F E6 84 83 00 2C 10 27 :77
2D20 00 03 16 05 57 DC 16 C3 00 01 DD 16 DC 1A 83 00 :97
2D30 01 10 26 00 0A DC 04 83 00 03 DD 04 16 00 21 DC :9B
2D40 1A 83 00 02 10 26 00 0A DC 04 83 00 02 DD 04 16 :3B
2D50 00 0E CC 36 06 9E 04 ED 84 DC 04 C3 00 02 DD 04 :AF
2D60 DC 08 83 00 04 DD 08 DC 1A 9E 08 ED 84 DC 24 9E :FB
2D70 08 ED 02 17 07 C3 9E 08 EC 84 83 00 01 10 26 00 :AB
2D80 19 CC 00 F7 9E 04 E7 84 DC 04 C3 00 03 DD 04 9E :0E
2D90 08 EC 02 9E 04 ED 1E 16 00 2A 9E 08 EC 84 83 00 :7C
2DA0 02 10 26 00 11 E6 03 86 9E 9E 04 ED 81 CC E7 84 :9D
2DB0 ED 81 9F 04 20 0E CC E7 D1 9E 04 ED 84 DC 04 C3 :79
2DC0 00 02 DD 04 DC 08 C3 00 04 DD 08 16 F4 BE 30 8C :F7
2DD0 02 20 06 43 50 4F 4B 45 00 34 10 17 05 F4 32 62 :82
2DE0 10 27 00 76 17 0B 7F DC 1A 83 00 02 10 27 00 03 :03
2DF0 16 04 89 DC 24 DD 34 CC 00 9E 9E 04 E7 1E 17 06 :E2
-------------------------------------------------------
Sum: 60 4F 80 ED 13 AE 3C 2B 52 98 86 68 22 43 2C A7 :54
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2E00 C1 DC 16 C3 00 01 DD 16 17 07 2E 9E 16 4F E6 84 :23
2E10 83 00 23 10 26 00 14 CC ED 81 9E 04 ED 84 DC 16 :2F
2E20 C3 00 01 DD 16 17 06 9A 16 00 07 CC E7 80 9E 04 :60
2E30 ED 84 DC 04 C3 00 02 DD 04 9E 16 4F E6 84 83 00 :E7
2E40 2C 27 BE CC 00 9F 9E 04 E7 84 DC 34 9E 04 E7 01 :23
2E50 DC 04 C3 00 02 DD 04 16 F4 32 30 8C 02 20 05 43 :E8
2E60 41 4C 4C 00 34 10 17 05 69 32 62 10 27 00 2E 17 :B2
2E70 06 C7 DC 1A 83 00 01 10 26 00 0A CC 00 BD 9E 04 :B2
2E80 E7 1D 16 F4 07 CC 1F 01 9E 04 ED 84 CC AD 84 9E :AF
2E90 04 ED 02 DC 04 C3 00 04 DD 04 16 F3 EF 30 8C 02 :31
2EA0 20 05 43 4F 44 45 00 34 10 17 05 26 32 62 10 27 :91
2EB0 00 60 DC 16 83 00 01 DD 16 DC 16 C3 00 01 DD 16 :72
2EC0 17 05 FF 17 0A A0 DC 1A 83 00 01 10 27 00 03 16 :A6
2ED0 03 AA 9E 16 4F E6 84 83 00 23 10 26 00 17 DC 04 :ED
2EE0 83 00 01 DD 04 DC 24 9E 04 ED 1E DC 16 C3 00 01 :C8
2EF0 DD 16 16 00 0D DC 04 83 00 02 DD 04 DC 24 9E 04 :FE
-------------------------------------------------------
Sum: C8 D2 AA D9 F4 B6 5B 5C B0 1B 8B CF 9D F6 15 F9 :44
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2F00 E7 1F 17 05 BD 9E 16 4F E6 84 83 00 2C 27 AA 16 :E2
2F10 F3 7A 30 8C 02 20 04 45 4E 44 00 34 10 17 04 B2 :37
2F20 32 62 10 27 00 26 CC 10 DE 9E 04 ED 84 CC FE 35 :BD
2F30 9E 04 ED 02 CC FF 00 9E 04 ED 04 CC 00 00 9E 04 :5D
2F40 E7 06 DC 04 C3 00 05 DD 04 16 F3 40 9E 16 4F E6 :A8
2F50 84 34 06 17 05 02 32 62 10 27 02 9C 17 03 34 9E :31
2F60 16 4F E6 84 83 00 5B 10 26 00 06 17 F6 E0 16 F3 :DF
2F70 1B 9E 16 4F E6 84 83 00 3B 10 26 00 6F DC 22 83 :6C
2F80 00 03 10 26 00 10 CC 00 00 9E 18 ED 84 DC 04 9E :BA
2F90 18 ED 02 16 F2 F6 9E 18 4F E6 01 83 00 00 10 26 :AA
2FA0 00 1B BD 4D 17 BD 4D 0B 44 4F 55 42 4C 45 20 44 :70
2FB0 45 46 20 45 52 52 4F 52 2E 00 16 10 8B 9E 18 EC :B6
2FC0 02 DD 36 CC 00 00 9E 18 E7 01 9E 36 EC 1E DD 2E :68
2FD0 DC 04 93 36 9E 36 ED 1E DC 2E DD 36 DC 36 83 00 :3A
2FE0 00 26 E7 DC 04 9E 18 ED 02 16 F2 A0 DC 22 83 00 :BB
2FF0 03 10 26 00 20 CC 01 00 9E 18 ED 84 DC 0C 9E 18 :EB
-------------------------------------------------------
Sum: 84 8E E7 54 D9 1E A5 29 AF D0 8A 32 B5 20 D2 35 :29
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3000 ED 02 DC 0C C3 00 02 DD 0C DC 0C 83 01 02 10 26 :29
3010 00 03 16 10 A6 9E 18 EC 02 DD 38 9E 16 4F E6 84 :F5
3020 83 00 3A 10 26 00 58 DC 16 C3 00 01 DD 16 17 05 :10
3030 08 DC 1A 83 00 01 10 26 00 08 CC 00 E7 DD 3A 16 :A0
3040 00 AE CC 00 9E 9E 04 E7 84 DC 38 9E 04 E7 01 CC :8F
3050 30 8B 9E 04 ED 02 CC 36 10 9E 04 ED 04 DC 04 C3 :94
3060 00 06 DD 04 DC 16 C3 00 02 DD 16 17 04 CB CC E7 :2A
3070 D1 9E 04 ED 84 DC 04 C3 00 02 DD 04 16 F2 0D 9E :1D
3080 16 4F E6 84 83 00 28 10 26 01 40 DC 16 C3 00 01 :A7
3090 DD 16 17 04 A4 DC 1A 83 00 01 10 26 00 0E CC 00 :3C
30A0 ED DD 3A DC 24 58 49 DD 24 16 00 44 CC 58 49 9E :0B
30B0 04 ED 84 CC 00 9E 9E 04 E7 02 DC 38 9E 04 E7 03 :0A
30C0 CC 30 8B 9E 04 ED 04 CC 36 10 9E 04 ED 06 DC 04 :A1
30D0 C3 00 08 DD 04 DC 16 C3 00 02 DD 16 17 04 5A CC :97
30E0 ED D1 9E 04 ED 84 DC 04 C3 00 02 DD 04 16 F1 9C :FA
30F0 DC 04 83 00 03 DD 04 DC 08 83 00 02 DD 08 DC 24 :95
-------------------------------------------------------
Sum: B5 F2 00 53 BD 2D 3C 8E EC 8C E8 3F 62 19 24 0B :F7
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3100 9E 08 ED 84 DC 16 C3 00 02 DD 16 17 04 2B CC 00 :D3
3110 9E 9E 04 E7 84 DC 38 9E 04 E7 01 DC 3A 9E 04 E7 :E8
3120 02 9E 08 EC 84 DD 24 DC 08 C3 00 02 DD 08 DC 24 :A7
3130 83 00 00 10 26 00 11 CC 00 84 9E 04 E7 03 DC 04 :86
3140 C3 00 04 DD 04 16 F1 44 DC 24 36 06 CC 00 10 BD :C8
3150 4D A1 36 06 DC 24 36 06 CC FF EF BD 4D A9 A4 C0 :37
3160 E4 C0 10 27 00 19 DC 24 36 06 CC 00 1F A4 C0 E4 :63
3170 C0 9E 04 E7 03 DC 04 C3 00 04 DD 04 16 F1 0D DC :C4
3180 24 36 06 CC 00 80 BD 4D A1 36 06 DC 24 36 06 CC :9B
3190 FF 7F BD 4D A9 A4 C0 E4 C0 10 27 00 17 CC 00 88 :DB
31A0 9E 04 E7 03 DC 24 9E 04 E7 04 DC 04 C3 00 05 DD :9E
31B0 04 16 F0 D8 CC 00 89 9E 04 E7 03 DC 24 9E 04 ED :52
31C0 04 DC 04 C3 00 06 DD 04 16 F0 C1 9E 16 4F E6 84 :C2
31D0 83 00 3D 10 26 00 21 DC 16 C3 00 01 DD 16 17 03 :DA
31E0 58 CC 00 DD 9E 04 E7 84 DC 38 9E 04 E7 01 DC 04 :8C
31F0 C3 00 02 DD 04 16 F0 94 9E 16 4F E6 84 83 00 25 :55
-------------------------------------------------------
Sum: DC BA 24 D9 06 66 B0 42 DE 6A 3D 05 D0 9B F1 1A :F1
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3200 10 26 00 7B DC 16 C3 00 01 DD 16 17 07 58 9E 16 :81
3210 4F E6 84 36 06 CC 00 3D BD 4D C1 36 06 DC 1A 36 :31
3220 06 CC 00 01 BD 4D C1 AA C0 EA C0 10 27 00 03 16 :02
3230 00 4A DC 04 83 00 03 DD 04 DC 24 58 49 DD 3C DC :27
3240 16 C3 00 01 DD 16 17 02 F0 DC 3C 83 00 10 10 2C :BD
3250 00 13 CC ED 60 D3 3C 9E 04 ED 84 DC 04 C3 00 02 :F3
3260 DD 04 16 00 14 CC ED E8 9E 04 ED 84 DC 3C 9E 04 :79
3270 E7 02 DC 04 C3 00 03 DD 04 16 F0 10 BD 4D 17 BD :64
3280 4D 0B 53 59 4E 54 41 58 20 45 52 52 4F 52 2E 00 :17
3290 16 0D B5 17 02 2C 9E 16 4F E6 84 34 06 17 01 B8 :94
32A0 32 62 83 00 00 12 12 12 10 26 00 03 16 FF CD CC :34
32B0 00 00 DD 1C CC 00 07 36 06 DC 1C 16 00 13 DC 1C :21
32C0 16 9E 0A 31 AB CC 00 20 E7 A4 DC 1C C3 00 01 DD :A4
32D0 1C A3 C4 2F E9 33 42 CC 00 00 DD 22 9E 16 4F E6 :C4
32E0 84 34 06 17 01 72 32 62 36 06 9E 16 4F E6 84 34 :B9
32F0 06 17 01 47 32 62 AA C0 EA C0 10 27 00 2F DC 22 :71
-------------------------------------------------------
Sum: 8A 04 5B EF 19 49 E0 ED A4 6A B1 C2 35 13 44 E6 :FA
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3300 83 00 08 10 2C 00 1C DC 22 10 9E 0A 31 AB 9E 16 :29
3310 4F E6 84 34 06 17 01 7D 32 62 E7 A4 DC 22 C3 00 :68
3320 01 DD 22 DC 16 C3 00 01 DD 16 16 FF AF CC 00 00 :39
3330 DD 22 DC 06 DD 18 9E 18 4F E6 84 83 00 FF 10 26 :FD
3340 00 48 CC 00 00 DD 1C CC 00 03 36 06 DC 1C 16 00 :26
3350 1D DC 1C C3 00 02 58 49 10 9E 18 31 AB DC 1C 9E :B3
3360 0A 58 49 EC 8B ED A4 DC 1C C3 00 01 DD 1C A3 C4 :CF
3370 2F DF 33 42 CC 00 00 9E 18 E7 84 CC 00 FF 9E 18 :F1
3380 E7 0C CC 00 03 DD 22 16 00 3C CC 00 00 DD 1C DC :B4
3390 1C C3 00 04 9E 18 E6 8B 4F 36 06 DC 1C 9E 0A E6 :1B
33A0 8B 4F BD 4D C1 10 27 00 0A DC 18 C3 00 0C DD 18 :9E
33B0 16 00 13 DC 1C C3 00 01 DD 1C DC 1C 83 00 08 26 :B7
33C0 CE CC 00 01 DD 22 DC 22 83 00 00 10 2F FF 67 16 :D6
33D0 00 F0 EC 62 DD 2E CC 00 00 DD 1C DC 1C 9E 2E E6 :B8
33E0 8B 4F 83 00 00 10 27 00 2E DC 1C 9E 2E E6 8B 4F :46
33F0 36 06 DC 1C 9E 16 E6 8B 4F 34 06 17 00 97 32 62 :24
-------------------------------------------------------
Sum: 39 6F D5 C3 52 FC B7 50 FA 10 F5 90 38 4C 41 63 :4C
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3400 BD 4D C1 10 27 00 06 CC 00 00 DD 2E 39 DC 1C C3 :D3
3410 00 01 DD 1C 16 FF C4 DC 16 D3 1C DD 16 17 00 A2 :60
3420 CC 00 01 DD 2E 39 CC 00 BD 9E 04 E7 84 DC 04 C3 :4A
3430 00 03 DD 04 EC 62 9E 04 ED 1E 39 EC 62 36 06 CC :6E
3440 00 30 BD 4D 99 36 06 EC 62 36 06 CC 00 39 BD 4D :A8
3450 B1 A4 C0 E4 C0 DD 2E 39 EC 62 36 06 CC 00 41 BD :51
3460 4D 99 36 06 EC 62 36 06 CC 00 5A BD 4D B1 A4 C0 :F1
3470 E4 C0 36 06 EC 62 36 06 CC 00 61 BD 4D 99 36 06 :76
3480 EC 62 36 06 CC 00 7A BD 4D B1 A4 C0 E4 C0 AA C0 :FD
3490 EA C0 DD 2E 39 EC 62 36 06 CC 00 60 BD 4D A9 36 :8D
34A0 06 EC 62 36 06 CC 00 7B BD 4D A1 A4 C0 E4 C0 10 :9A
34B0 27 00 0A EC 62 83 00 20 DD 40 16 00 04 EC 62 DD :84
34C0 40 39 9E 16 4F E6 84 83 00 20 10 26 00 0A DC 16 :BB
34D0 C3 00 01 DD 16 16 FF EA 39 DC 22 83 00 03 10 26 :A9
34E0 00 15 CC 00 00 9E 04 ED 1E CC 00 01 9E 18 ED 84 :82
34F0 DC 04 9E 18 ED 02 39 9E 18 4F E6 01 83 00 10 :3D
-------------------------------------------------------
Sum: 4D DE ED AB 47 48 70 63 02 48 A0 99 21 8A 4C 77 :16
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3500 26 00 0B 9E 18 EC 02 93 04 9E 04 ED 1E 39 9E 18 :08
3510 EC 02 9E 04 ED 1E DC 04 9E 18 ED 02 39 17 00 19 :89
3520 9E 16 4F E6 84 83 00 29 10 27 00 03 16 0A FF DC :4E
3530 16 C3 00 01 DD 16 16 FF 89 17 FF 86 17 00 A3 9E :5F
3540 16 4F E6 84 DD 36 30 8C 02 20 04 41 4E 44 00 34 :CB
3550 10 17 FE 7E 32 62 36 06 30 8C 02 20 03 4F 52 00 :F5
3560 34 10 17 FE 6D 32 62 AA C0 EA C0 10 27 00 70 DC :F1
3570 08 83 00 02 DD 08 DC 36 9E 08 ED 84 CC 36 06 9E :41
3580 04 ED 84 DC 04 C3 00 02 DD 04 17 00 55 9E 08 EC :F9
3590 81 9F 08 C1 41 27 09 C1 61 27 05 CC 00 8A 20 03 :21
35A0 CC 00 84 DD 2E 9E 16 4F E6 84 DD 36 12 12 12 12 :23
35B0 DC 2E C3 00 20 9E 04 E7 84 CC 00 C0 9E 04 E7 01 :10
35C0 DC 2E C3 00 60 9E 04 E7 02 CC 00 C0 9E 04 E7 03 :D0
35D0 DC 04 C3 00 04 DD 04 CC 00 05 DD 1A 16 FF 67 16 :E2
35E0 FE E0 17 FE DD 17 00 E7 30 8C 02 20 03 3C 3E 00 :29
35F0 34 10 17 FD DD 32 62 10 27 00 08 CC 4D C1 DD 2C :EB
-------------------------------------------------------
Sum: 3F B0 7A 00 70 5F 25 D4 CC 6A 83 F5 D1 61 92 A0 :43
```

リスト6　拡張Kコンパイラ(KV3.6)

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3600 16 00 B1 30 8C 02 20 03 3C 3D 00 34 10 17 FD C2 :0B
3610 32 62 10 27 00 08 CC 4D B1 DD 2C 16 00 66 30 8C :DE
3620 02 20 03 3E 3D 00 34 10 17 FD A7 32 62 10 27 00 :6A
3630 08 CC 4D 99 DD 2C 16 00 4B 30 8C 02 20 02 3D 00 :41
3640 34 10 17 FD 8D 32 62 10 27 00 08 CC 4D B9 DD 2C :93
3650 16 00 31 30 8C 02 20 02 3E 00 34 10 17 FD 73 32 :62
3660 62 10 27 00 08 CC 4D A9 DD 2C 16 00 17 30 8C 02 :57
3670 20 02 3C 00 34 10 17 FD 59 32 62 10 27 00 4D CC :F3
3680 4D A1 DD 2C DC 08 83 00 02 DD 08 DC 2C 9E 08 ED :E0
3690 84 CC 36 06 9E 04 ED 84 DC 04 C3 00 02 DD 04 17 :3C
36A0 00 2D 9E 08 EC 84 DD 2C DC 08 C3 00 02 DD 08 CC :A6
36B0 00 BD 9E 04 E7 84 DC 04 C3 00 03 DD 04 DC 2C 9E :F7
36C0 04 ED 1E DC 1A DD 42 CC 00 04 DD 1A 16 FD F3 17 :08
36D0 FD F0 17 01 21 9E 16 4F E6 84 36 06 CC 00 2B BD :83
36E0 4D B9 36 06 9E 16 4F E6 84 36 06 CC 00 2D BD 4D :EE
36F0 B9 AA C0 EA C0 10 27 00 FA DC 08 83 00 02 DD 08 :4C
---------------------------------------------------------
Sum: F6 07 06 66 E1 FB 13 CD CB 28 C5 92 4A D5 B2 11 :51
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3700 9E 16 4F E6 84 9E 08 ED 84 DC 16 C3 00 01 DD 16 :2D
3710 CC 36 06 9E 04 ED 84 DC 04 C3 00 02 DD 04 17 00 :B8
3720 D5 9E 08 EC 84 DD 2E DC 08 C3 00 02 DD 08 DC 2E :8E
3730 83 00 2B 10 26 00 51 DC 1A 83 00 01 10 26 00 17 :FC
3740 DC 04 83 00 02 DD 04 CC 00 C3 9E 04 E7 1D DC 24 :7B
3750 9E 04 ED 1E 16 00 2E DC 1A 83 00 02 10 26 00 17 :B9
3760 DC 04 83 00 02 DD 04 CC 00 D3 9E 04 E7 1E DC 24 :8C
3770 9E 04 E7 1F 16 00 0E CC E3 C1 9E 04 ED 84 DC 04 :2F
3780 C3 00 02 DD 04 16 00 63 DC 1A 83 00 01 10 26 00 :CF
3790 17 DC 04 83 00 02 DD 04 CC 00 83 9E 04 E7 1D DC :2E
37A0 24 9E 04 ED 1E 16 00 43 DC 1A 83 00 02 10 26 00 :DB
37B0 17 DC 04 83 00 02 DD 04 CC 00 93 9E 04 E7 1E DC :3F
37C0 24 9E 04 E7 1F 16 00 23 CC A3 C1 9E 04 ED 84 CC :14
37D0 53 43 9E 04 ED 02 CC C3 00 9E 04 ED 04 CC 00 01 :16
37E0 9E 04 E7 06 DC 04 C3 00 07 DD 04 CC 00 05 DD 1A :E2
37F0 16 FE E2 16 FC CC 17 FC C9 17 01 6A 9E 16 4F E6 :1B
---------------------------------------------------------
Sum: F6 33 DB 94 68 3A AF 51 93 28 D6 D3 46 DA 9B 43 :9C
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3800 84 36 06 CC 00 2A BD 4D B9 36 06 9E 16 4F E6 84 :22
3810 36 06 CC 00 2F BD 4D B9 AA C0 EA C0 10 27 01 43 :89
3820 DC 08 83 00 02 DD 08 9E 16 4F E6 84 9E 08 ED 84 :D2
3830 DC 16 C3 00 01 DD 16 CC 36 06 9E 04 ED 84 DC 04 :A4
3840 C3 00 02 DD 04 17 01 1E 9E 08 EC 84 DD 2E DC 08 :E1
3850 C3 00 02 DD 08 DC 2E 83 00 2A 10 26 00 80 DC 1A :0D
3860 36 06 CC 00 01 BD 4D B9 36 06 DC 24 36 06 CC 00 :10
3870 02 BD 4D B9 36 06 DC 24 36 06 CC 00 04 BD 4D B9 :D0
3880 AA C0 EA C0 36 06 DC 24 36 06 CC 00 08 BD 4D B9 :23
3890 AA C0 EA C0 36 06 DC 24 36 06 CC 00 10 BD 4D B9 :2B
38A0 AA C0 EA C0 A4 C0 E4 C0 10 27 00 25 DC 04 83 00 :DB
38B0 05 DD 04 CC 58 49 9E 04 ED 84 DC 04 C3 00 02 DD :E8
38C0 04 DC 24 47 56 DD 24 DC 24 83 00 01 26 E5 16 00 :47
38D0 0A CC 4D 1F 34 06 17 FB 4D 32 62 16 00 7D DC 1A :F8
38E0 36 06 CC 00 01 BD 4D B9 36 06 DC 24 36 06 CC 00 :10
38F0 02 BD 4D B9 36 06 DC 24 36 06 CC 00 04 BD 4D B9 :D0
---------------------------------------------------------
Sum: 79 A5 81 6A 9E 12 1E AE FF 01 96 18 DF 16 AB 4C :1F
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3900 AA C0 EA C0 36 06 DC 24 36 06 CC 00 08 BD 4D B9 :23
3910 AA C0 EA C0 36 06 DC 24 36 06 CC 00 10 BD 4D B9 :2B
3920 AA C0 EA C0 A4 C0 E4 C0 10 27 00 25 DC 04 83 00 :DB
3930 05 DD 04 CC 47 56 9E 04 ED 84 DC 04 C3 00 02 DD :E4
3940 04 DC 24 47 56 DD 24 DC 24 83 00 01 26 E5 16 00 :47
3950 0A CC 4D 3B 34 06 17 FA CD 32 62 CC 00 05 DD 1A :D2
3960 16 FE 99 16 FB 5C 30 8C 02 20 06 49 4E 50 55 54 :8E
3970 00 34 10 17 FA 5C 32 62 10 27 00 12 CC 4D E2 34 :BD
3980 06 17 FA A2 32 62 CC 00 03 DD 1A 16 FB 34 30 8C :14
3990 02 20 04 47 45 54 00 34 10 17 FA 36 32 62 10 27 :5C
39A0 00 12 CC 4D D6 34 06 17 FA 7C 32 62 CC 00 03 DD :08
39B0 1A 16 FB 0E 30 8C 02 20 05 41 42 53 28 00 34 10 :5E
39C0 17 FA 0F 32 62 10 27 00 15 17 FB 51 CC 4D 95 34 :45
39D0 06 17 FA 52 32 62 CC 00 03 DD 1A 16 FA E4 30 8C :73
39E0 02 20 05 4E 4F 54 28 00 34 10 17 F9 E5 32 62 10 :1D
39F0 27 00 15 17 FB 27 CC 4D CD 34 06 17 FA 28 32 62 :62
---------------------------------------------------------
Sum: 8F 87 C4 E8 31 20 92 88 97 9C 96 C9 BD 26 19 C3 :7E
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3A00 CC 00 03 DD 1A 16 FA BA 30 8C 02 20 05 41 44 52 :4A
3A10 28 00 34 10 17 F9 BB 32 62 10 27 00 2C 17 FA FD :3C
3A20 DC 1A 83 00 02 10 27 00 03 16 FB 50 CC 00 03 DD :BF
3A30 1A CC 00 C6 9E 04 E7 1E CC 1F B8 9E 04 ED 84 DC :E5
3A40 04 C3 00 02 DD 04 16 FA 79 30 8C 02 20 06 48 49 :A8
3A50 47 48 28 00 34 10 17 F9 79 32 62 10 27 00 20 17 :86
3A60 FA BB CC 1F 89 9E 04 ED 84 CC 00 4F 9E 04 E7 02 :E2
3A70 DC 04 C3 00 03 DD 04 CC 00 03 DD 1A 16 FA 43 30 :D0
3A80 8C 02 20 05 4C 4F 57 28 00 34 10 17 F9 44 32 62 :F9
3A90 10 27 00 19 17 FA 86 CC 00 4F 9E 04 E7 84 DC 04 :EF
3AA0 C3 00 01 DD 04 CC 00 03 DD 1A 16 FA 15 30 8C 02 :4E
3AB0 20 05 53 45 58 28 00 34 10 17 F9 16 32 62 10 27 :72
3AC0 00 19 17 FA 58 CC 00 1D 9E 04 E7 84 DC 04 C3 00 :1B
3AD0 01 DD 04 CC 00 03 DD 1A 16 F9 E7 30 8C 02 20 05 :81
3AE0 53 47 4E 28 00 34 10 17 F8 E8 32 62 10 27 00 4A :60
3AF0 17 FA 2A CC 00 03 DD 1A CC 83 00 9E 04 ED 84 CC :2F
---------------------------------------------------------
Sum: F5 15 78 CE 85 F5 9F 49 3C 1E 61 68 9F BD 68 44 :DD
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3B00 00 27 9E 04 ED 02 CC 0A 2A 9E 04 ED 04 CC 05 CC :E8
3B10 9E 04 ED 06 CC FF FF 9E 04 ED 08 CC 20 03 9E 04 :87
3B20 ED 0A CC CC 00 9E 04 ED 0C CC 00 01 9E 04 E7 0E :8E
3B30 DC 04 C3 00 0F DD 04 16 F9 88 9E 16 4F E6 84 83 :1A
3B40 00 2D 10 26 00 49 DC 16 C3 00 01 DD 16 17 FE 16 :80
3B50 DC 1A 83 00 01 10 26 00 12 DC 24 53 43 C3 00 01 :1C
3B60 DD 24 DC 24 9E 04 ED 1E 16 00 21 CC 53 43 9E 04 :E9
3B70 ED 84 CC C3 00 9E 04 ED 02 CC 00 01 9E 04 E7 04 :EB
3B80 DC 04 C3 00 05 DD 04 CC 00 03 DD 1A 16 F9 33 30 :C1
3B90 8C 02 20 06 50 45 45 4B 28 00 34 10 17 F8 33 32 :B9
3BA0 62 10 27 00 66 17 F9 75 DC 1A 83 00 01 10 26 00 :34
3BB0 17 DC 04 C3 00 01 DD 04 CC 4F F6 9E 04 ED 1C DC :34
3BC0 24 9E 04 ED 1E 16 00 3D DC 1A 83 00 02 10 26 00 :D5
3BD0 18 C6 9E 9E 04 E7 1E C6 4F E7 80 CC E6 84 ED 81 :43
3BE0 9F 04 12 12 12 12 16 00 1C CC 1F 01 9E 04 ED 84 :1C
3BF0 CC 4F E6 9E 04 ED 02 CC 00 84 9E 04 E7 04 DÇ 04 :4F
---------------------------------------------------------
Sum: 95 D1 FD E7 5A AD 1B 2B 37 44 3A 66 FA 64 15 C7 :EC
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3C00 C3 00 05 DD 04 CC 00 03 DD 1A 39 9E 16 4F E6 84 :15
3C10 83 00 25 10 26 00 5B DC 16 C3 00 01 DD 16 17 FD :F6
3C20 45 DC 1A 83 00 01 10 27 00 03 16 F6 4F DC 04 83 :B7
3C30 00 01 DD 04 CC 00 EC 9E 04 E7 1E DC 24 58 49 DD :BF
3C40 24 DC 24 83 00 10 10 2C 00 0C CC 00 60 D3 24 9E :C0
3C50 04 E7 1F 16 00 14 CC 00 E8 9E 04 E7 1F DC 24 9E :2E
3C60 04 E7 84 DC 04 C3 00 01 DD 04 CC 00 06 DD 1A 16 :D3
3C70 F8 50 30 8C 02 20 05 52 4E 44 28 00 34 10 17 F7 :89
3C80 51 32 62 10 27 00 15 17 F8 93 CC 4F 09 34 06 17 :48
3C90 F7 94 32 62 CC 00 03 DD 1A 16 F8 26 9E 16 4F E6 :02
3CA0 84 83 00 28 10 26 00 23 DC 16 C3 00 01 DD 16 17 :48
3CB0 F8 87 9E 16 4F E6 84 83 00 29 10 27 00 03 16 03 :EB
3CC0 6D DC 16 C3 00 01 DD 16 16 F7 F7 9E 16 4F E6 84 :87
3CD0 83 00 24 10 26 00 78 CC 00 00 DD 2E CC 00 00 DD :D5
3CE0 24 DC 2E 83 00 09 10 2F 00 07 DC 2E 83 00 07 DD :71
3CF0 2E DC 24 58 49 58 49 58 49 58 49 D3 2E DD 24 DC :90
---------------------------------------------------------
Sum: B5 3B D6 D3 BD 42 82 26 57 F7 C1 C1 5A 8B 55 5B :A5
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3D00 16 C3 00 01 DD 16 9E 16 4F E6 84 83 00 30 DD 2E :F8
3D10 DC 2E 36 06 CC 00 09 BD 4D A9 36 06 DC 2E 36 06 :50
3D20 CC 00 11 BD 4D A1 A4 C0 E4 C0 36 06 DC 2E 36 06 :12
3D30 CC 00 00 BD 4D A1 AA C0 EA C0 36 06 DC 2E 36 06 :0D
3D40 CC 00 16 BD 4D A9 AA C0 EA C0 27 95 16 00 45 9E :5E
3D50 16 4F E6 84 34 06 17 F6 E2 32 62 10 27 00 51 CC :E0
3D60 00 00 DD 24 DC 24 36 06 CC 00 0A BD 4D 1F 36 06 :78
3D70 9E 16 4F E6 84 E3 C1 83 00 30 DD 24 DC 16 C3 00 :7A
3D80 01 DD 16 9E 16 4F E6 84 34 06 17 F6 AE 32 62 83 :6D
3D90 00 00 26 D0 CC 00 CC 9E 04 E7 84 DC 04 C3 00 03 :41
3DA0 DD 04 DC 24 9E 04 ED 1E CC 00 01 DD 1A 16 F7 12 :71
3DB0 9E 16 4F E6 84 83 00 27 10 26 00 11 9E 16 4F E6 :47
3DC0 01 DD 24 DC 16 C3 00 03 DD 16 16 FF C7 9E 16 4F :8C
3DD0 E6 84 34 06 17 F6 B1 32 62 10 27 02 51 17 F4 B3 :0E
3DE0 9E 18 4F E6 84 83 00 03 10 26 00 09 9E 18 EC 02 :D8
3DF0 DD 24 16 FF 9F 9E 16 4F E6 84 83 00 5B 10 26 00 :36
---------------------------------------------------------
Sum: E8 EA 93 0B 78 BE E3 80 4B 14 F2 E5 75 ED D2 32 :A5
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3E00 0E CC 00 17 DD 2C 17 E8 45 CC 00 06 DD 1A 39 DC :1C
3E10 22 83 00 03 10 26 00 20 CC 01 00 9E 18 ED 84 DC :CE
3E20 0C 9E 18 ED 02 DC 0C C3 00 02 DD 0C DC 0C 83 01 :B3
3E30 02 10 26 00 03 16 02 83 9E 18 EC 02 DD 2E 9E 16 :39
3E40 4F E6 84 83 00 3A 10 26 00 91 DC 16 C3 00 01 DD :D0
3E50 16 DC 08 83 00 02 DD 08 DC 2E 9E 08 ED 84 17 F6 :92
3E60 D8 9E 08 EC 84 DD 2E DC 08 C3 00 02 DD 08 9E 16 :3B
3E70 4F E6 84 83 00 29 10 27 00 03 16 01 B1 DC 16 C3 :1C
3E80 00 01 DD 16 DC 1A 83 00 01 10 26 00 26 CC 00 03 :99
3E90 DD 1A DC 04 83 00 03 DD 04 CC 00 9E 9E 04 E7 84 :B5
3EA0 DC 2E 9E 04 E7 01 CC 4F E6 9E 04 ED 02 16 00 A0 :DC
3EB0 16 00 28 CC 00 9E 9E 04 E7 84 DC 2E 9E 04 E7 01 :49
3EC0 CC E6 8B 9E 04 ED 02 CC 00 4F 9E 04 E7 04 DC 04 :56
3ED0 C3 00 05 DD 04 CC 00 03 DD 1A 39 9E 16 4F E6 84 :15
3EE0 83 00 28 10 26 01 29 DC 16 C3 00 01 DD 16 DC 08 :98
3EF0 83 00 02 DD 08 DC 2E 9E 08 ED 84 17 F6 3B 9E 08 :79
---------------------------------------------------------
Sum: 2E 72 8F CE F2 D5 99 F8 60 83 BA 46 20 37 B4 3B :7E
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3F00 EC 84 DD 2E DC 08 C3 00 02 DD 08 9E 16 4F E6 84 :76
3F10 83 00 29 10 27 00 03 16 01 14 DC 16 C3 00 01 DD :A4
3F20 16 DC 1A 83 00 01 10 26 00 BE CC 00 03 DD 1A DC :26
3F30 04 83 00 04 DD 04 CC 00 9E 9E 04 E7 01 DC 2E 9E :08
3F40 04 E7 02 CC 00 EC 9E 04 E7 03 DC 24 58 49 DD 24 :D3
3F50 DC 24 83 00 00 10 26 00 0F CC 00 84 9E 04 E7 04 :A5
3F60 DC 04 C3 00 05 DD 04 39 DC 24 36 06 CC 00 10 BD :97
3F70 4D A1 36 06 DC 24 36 06 CC FF EF BD 4D A9 A4 C0 :37
3F80 E4 C0 10 27 00 17 DC 24 36 06 CC 00 1F A4 C0 E4 :61
3F90 C0 9E 04 E7 04 DC 04 C3 00 05 DD 04 39 DC 24 36 :45
3FA0 06 CC 00 80 BD 4D A1 36 06 DC 24 36 06 CC FF 7F :BF
3FB0 BD 4D A9 A4 C0 E4 C0 10 27 00 15 CC 00 88 9E 04 :FD
3FC0 E7 04 DC 24 9E 04 E7 05 DC 04 C3 00 06 DD 04 39 :3C
3FD0 CC 00 89 9E 04 E7 04 DC 04 C3 00 07 DD 04 DC 24 :6D
3FE0 9E 04 ED 1E 39 16 00 28 CC 00 9E 9E 04 E7 84 DC :77
3FF0 2E 9E 04 E7 01 CC 58 49 9E 04 ED 02 CC EC 8B 9E :97
---------------------------------------------------------
Sum: 7B B0 B1 90 1E FB 24 FE EC F1 E5 B3 FD 86 17 F4 :A7
```

リスト 6　拡張Kコンパイラ(KV3.6)

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4000 04 ED 04 DC 04 C3 00 06 DD 04 CC 00 03 DD 1A 39 :7E
4010 CC 00 DC 9E 04 E7 84 DC 2E 9E 04 E7 01 DC 04 C3 :EC
4020 00 02 DD 04 CC 00 02 DD 1A DC 2E DD 24 39 BD 4D :F6
4030 17 BD 4D 0B 49 4C 4C 45 47 41 4C 20 45 58 50 52 :85
4040 45 53 53 49 4F 4E 2E 00 BD 4D 0B 20 41 54 20 4C :35
4050 49 4E 45 20 00 DC 12 BD 4E B0 BD 4D 17 BD 4D 17 :E7
4060 DC 00 DD 2E 9E 2E 4F E6 84 BD 4E DE DC 2E C3 00 :22
4070 01 DD 2E 9E 2E 4F E6 1F 83 00 0D 26 E7 CC 00 0A :9F
4080 BD 4E DE DC 16 93 00 10 27 00 07 DC 16 93 00 BD :EE
4090 4E B6 BD 4D 0B 5E 00 17 01 56 BD 4D 17 17 DF 69 :65
40A0 BD 4D 17 BD 4D 0B 49 4C 4C 45 47 41 4C 20 41 52 :E3
40B0 47 55 4D 45 4E 54 2E 00 16 FF 8D BD 4D 17 BD 4D :CB
40C0 0B 54 4F 4F 20 4D 41 4E 59 20 56 41 52 49 41 42 :27
40D0 4C 45 53 2E 00 16 FF 70 DC 1A 36 06 CC 00 04 BD :56
40E0 4D B9 36 06 DC 42 36 06 CC 00 03 BD 4D A1 A4 C0 :7A
40F0 E4 C0 10 27 00 8A DC 04 83 00 05 DD 04 DC 42 83 :4F
---------------------------------------------------------
Sum: E9 E2 94 93 F0 1C 10 01 8C 4D 99 5D BD FC 63 0F :09
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4100 00 01 10 26 00 10 CC 00 83 9E 04 E7 1D DC 24 9E :DA
4110 04 ED 1E 16 00 0D CC 00 93 9E 04 E7 1E DC 24 9E :D6
4120 04 E7 1F DC 2C 83 4D 99 10 26 00 08 CC 10 2D DD :9F
4130 2C 16 00 49 DC 2C 83 4D A9 10 26 00 08 CC 10 2F :55
4140 DD 2C 16 00 38 DC 2C 83 4D A1 10 26 00 08 CC 10 :EA
4150 2C DD 2C 16 00 27 DC 2C 83 4D B1 10 26 00 08 CC :05
4160 10 2E DD 2C 16 00 16 DC 2C 83 4D B9 10 26 00 08 :42
4170 CC 10 26 DD 2C 16 00 05 CC 10 27 DD 2C 16 00 05 :4D
4180 CC 10 27 DD 2C 39 DC 00 12 DD 16 9E 46 EC 84 DD :57
4190 44 9E 46 EC 02 DD 12 DC 44 83 00 00 10 26 00 08 :E6
41A0 CC 7F 0D 9E 16 ED 84 39 DC 46 C3 00 06 DD 46 DC :A0
41B0 46 93 44 10 2C 00 1A 9E 46 4F E6 84 9E 16 E7 84 :2F
41C0 DC 16 C3 00 01 DD 16 DC 46 C3 00 01 DD 46 16 FF :C7
41D0 DE CC 00 0D 9E 16 E7 1F DC 44 DD 46 DC 00 12 DD :7F
41E0 16 39 CC 00 33 DD 46 9E 46 EC 84 DD 46 39 39 39 :93
41F0 BD 4D 17 10 DE FE 35 7F 7E 4F EB 00 00 00 00 00 :79
---------------------------------------------------------
Sum: C8 5A F6 14 A2 B6 8A 41 F5 2A 6E E8 6A 5C 6B 8B :80
```

リスト 7　ファイル版への変更

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4180 CC 10 27 DD 2C 39 34 48 BD 4D 06 86 0D 10 9E 00 :12
4190 A7 A4 10 8E 00 00 EC 84 10 83 7F 0D 27 28 A6 80 :ED
41A0 81 0D 27 31 81 27 27 1E 80 30 2B F2 81 39 22 EE :6A
41B0 34 02 1F 20 34 06 58 49 58 49 E3 E1 58 49 EB E0 :21
41C0 89 00 1F 02 20 D8 34 20 10 9E 00 A6 80 A7 A0 81 :92
41D0 0D 26 F8 35 20 10 9F 12 9E 00 9F 16 35 C8 12 DD :80
41E0 16 39 CC 00 33 DD 46 9E 46 EC 84 DD 46 39 39 39 :93
41F0 BD 4D 17 10 DE FE 35 FF 00 00 00 00 00 00 00 00 :41
---------------------------------------------------------
Sum: 91 6F 77 03 32 29 ED 02 99 D3 B6 FF 08 62 3C E5 :70
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4FA0 20 D9 34 6F 4F 1F 8B C6 01 D7 BF BD D8 07 25 11 :C4
4FB0 0D C0 26 0D 8E 04 3D A6 80 26 FC 86 0D A7 82 20 :F3
4FC0 08 CC 7F 0D 8E 04 3D ED 84 8E 04 3D 35 EF 7D FC :0C
4FD0 84 26 01 4F 7F FD 05 39 E7 84 8D B9 C6 12 8D 02 :CC
---------------------------------------------------------
Sum: B9 8B DA D8 EA 24 0A 92 EC 0F 4C 39 E0 AF B1 2F :8F
```

4D09：20→39

リスト 8　メイン

```
10 CLEAR 300,&H1FFF
20 INPUT"HOT OR COLD";A$
30 IF A$="H" THEN 60
40 LOADM"RTF
50 LOADM"KF
60 INPUT"FILE NAME";A$
70 OPEN"I",#1,A$
80 IF PEEK(&H2DC)=0 THEN GOTO 120
90 PRINT"FILE IS OPENED"
100 EXEC&H2000
110 GOTO 130
120 PRINT"** NOT ASCII FILE **"
130 CLOSE#1
```

リスト 9　Kコンパイラ・バージョン・アップ(V1.3→V3.6)

Addr	old	new	Addr	old	new	Addr	old	new	Addr	old	new	Addr	old	new
2000	C6	34	2001	4C	7F	2002	1F	C6	2003	9B	4C	2004	CE	1F
2005	44	9B	2006	00	CE	2007	8D	4C	2008	07	00	2009	C6	10
200A	00	DF	200B	1F	FE	200C	9B	12	200D	7E	12	200E	94	12
200F	7C	12	2029	31	33	202B	33	36	2073	43	42	2082	44	43
208E	4A	4B	208F	FE	8E	20B8	12	34	20B9	12	7F	20BD	84	81
20C1	9E	12	20C2	04	12	20C4	02	81	20C8	9E	12	20C9	04	12
20CB	04	81	20CF	9E	12	20D0	04	12	20D2	06	81	20D5	00	10
20D6	9E	12	20D7	04	12	20D8	E7	ED	20D9	08	81	20DA	DC	CC
20DB	04	DF	20DC	C3	FE	20DD	00	ED	20DE	09	81	20DF	DD	9F
20F3	C6	10	20F4	00	DE	20FA	1F	FE	20FB	9B	35	2101	7E	FF
2102	94	00	2109	8D	00	2112	07	05	2257	83	A3	2258	4F	8D
2259	FF	FE	225A	DD	22	225B	1E	DD	225C	DC	1E	225D	1E	12
2288	BD	17	2289	4D	1F	228A	17	65	2871	F9	07	2D48	CC	DC
2D49	00	04	2D4A	9E	83	2D4B	9E	00	2D4C	04	02	2D4D	E7	DD
2D4E	1E	04	2DA5	CC	E6	2DA6	E7	03	2DA7	84	86	2DA9	04	9E
2DAA	ED	04	2DAB	84	ED	2DAC	DC	81	2DAD	04	CC	2DAE	C3	E7
2DAF	00	84	2DB0	02	ED	2DB1	DD	81	2DB2	04	9F	2DB3	16	04
2DB4	00	20	2F27	C6	10	2F28	00	DE	2F2E	1F	FE	2F2F	9B	35
2F35	7E	FF	2F36	94	00	2F3D	8D	00	2F46	07	05	3590	84	81
3591	DD	9F	3592	2E	08	3593	DC	C1	3594	08	41	3595	C3	27
3596	00	09	3597	02	C1	3598	DD	61	3599	08	27	359A	DC	05
359B	2E	CC	359C	83	00	359D	00	8A	359E	41	20	359F	10	03
35A0	26	CC	35A2	08	84	35A3	CC	DD	35A4	00	2E	35A5	84	9E
35A6	DD	16	35A7	2E	4F	35A8	16	E6	35A9	00	84	35AA	05	DD
35AB	CC	36	35AC	00	12	35AD	8A	12	35AE	DD	12	35AF	2E	12
3AB8	08	10	3BD1	CC	C6	3BD2	00	9E	3BD4	9E	04	3BD5	04	E7
3BD6	E7	1E	3BD7	1E	C6	3BD8	CC	4F	3BD9	EC	E7	3BDA	84	80
3BDB	9E	CC	3BDC	04	E6	3BDD	ED	84	3BDE	84	ED	3BDF	DC	81
3BE0	04	9F	3BE1	C3	04	3BE2	00	12	3BE3	02	12	3BE4	DD	12
3BE5	04	12	4097	BD	17	4098	4D	01	4099	17	56	409D	BD	17
409E	20	DF	409F	09	69	41F0	39	BD	41F1	35	4D	41F2	29	17
41F3	3D	10	41F4	24	DE	41F5	44	FE	41F6	44	35	41F7	20	7F
41F8	59	7E	41F9	3A	4F	41FA	36	EB	41FB	29	00	41FC	3D	00
41FD	53	00	41FE	28	00									

Kコンパイラやマシンをフルに活用！

IODOS9 アイオードスナイン

■Wonder Soft Sgn

このプログラムは，雑誌I/O，FM-8，7/8活研に掲載された，種々のユーティリティを活用するためのDOS（ディスク・オペレーティング・システム）です．名付けて，"IODOS 9" です．

リロケータブルで，BIOSを使うプログラムなら，なんでも（もちろん大きすぎてはいけません），コマンドとして登録して使えます．

IODOS9の特徴

❶F-BASICと互換性のあるディスク管理を行なう．
❷フルRAMで動作する．使うのは，ブートROMとサブシステムのコマンドのみ．
❸常駐コマンド（6種）の他に，非常駐コマンドがある．
❹ファイル名の拡張（ファイル・タイプ，ファイル・モード）ができる．

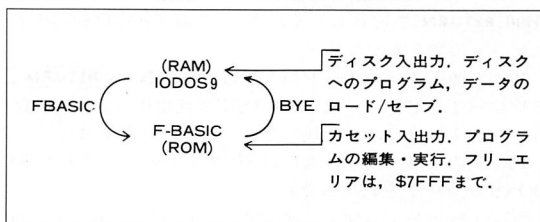
対象システム

機種：FM-7
装置：5インチ両面倍密度フロッピーディスク装置（1または2ドライブ）．
メモリ：$CD00～$EFFFを占有する．
F-BASIC：バージョン3.0

F-BASICとの互換性

❶IODOS 9上で，F-BASICのディスクの読み書きができる．F-BASIC上ではIODOS 9のディスクの**FILES**，**LOAD**，**SAVE**はできない（"Bad File Structure"となる）．
❷常駐コマンドはF-BASICに準拠している．
❸コマンドで入出力できるのは，BASICプログラムとマシン語ファイルである．ASCII形式ファイル，ランダムアクセス・ファイルは扱えない．
❹非常駐コマンドの**FBASIC**により，IODOS 9からF-BASICに制御を移せる．ロード済のF-BASICプログラムはそのまま残っている．F-BASICからIODOS 9に戻るには，拡張コマンド**BYE**を用いる．

注） F-BASICはROMモードで，フリーエリアは$7FFFまでとなるが，ディスク関係の命令は使えない（**図1**）．

図1　IODOS 9とF-BASIC

ファイル名の拡張

ファイル名は，F-BASICのファイル名の他に，ファイル・タイプ(*ft*)，ファイル・モード(*fm*)がつけられます．*ft*，*fm*はこの順に空白で区切って指定する方式です．空白を2個続けると，*ft*を省略できます．

fn：F-BASICのファイル名．8文字内の英字で始まる，英数字列．
ft：ファイルの形式を示す．3文字以内．省略可．
fm：ファイルのモードを示す．1文字．省略可．
- R：読み出しのみ可能．SAVEやKILLを禁止する．
- W：書き込みのみ可能．LOADを禁止する．

注）：すでに，*fn*のなかに空白を含んでいるファイルは，F-BASIC（DISKモード）で，名前を変更してから使ってください．このとき，F-BASICでファイルを書きかえる（再SAVE）と，ファイル属性やDATE情報が消されます．

非常駐コマンドの登録と実行

*ft*が**COM**で，*fm*が**R**であるファイルは，ファイル名(*fn*)を指定するだけで，ディスクから非常駐コマンド領域にロードされ，実行されます．*fn*，*ft*，*fm*とも全部大文字で指定してください．

IODOS 9では，ブート時にROMにあるBIOSを裏RAMにコピーするので，BIOSを使ったリロケータブルなプログラムなら，名前を換えるだけでコマンドとして使えます．

非常駐コマンドから，IODOS 9に戻るには，リターン命令（**RTS**など）を用います．IODOS 9からは，BREAKキー割り込み可状態で制御が渡されるので，BREAKキーを押すとIODOS 9に強制的に戻されます．

非常駐コマンド内では，レジスタ類を自由に使えます．スタックを変更してしまったときは，IODOSの先頭($D000)にジャンプしてください．

なお，非常駐コマンドは，$A000から$CBFFまでの間にロードされます．非常駐コマンドのパラメータは，非常駐コマンド名のあとに，1個以上の空白または，記号（引用符(")など）をつけて指定します．パラメータは，先頭の空白が除かれて，$EE60から$EE7Fまでに入ります．パラメータの最後には，ヌル(0)が入っています．

例1 ABC ␣ D, E, F RETURN $EE60 +0 +1 +2 +3 +4 +5
44 2C 45 2C 46 00
パラメータ
非常駐コマンド(大文字でも小字文でも良い)
例2 ABC "DEF" RETURN $EE60 +0 +1 +2 +3 +4 +5
22 44 45 46 22 00
パラメータ
非常駐コマンド

IODOS9の作り方

リスト1のダンプ・リストを打ち込んだあと，ドライブ0にフォーマット*[1]済のディスケットを入れ，**EXEC & H 1B00** RETURN を実行してください．これでIODOS9のディスケットのできあがりです．

初めて使うディスケットのときは，**DSKINI0** RETURN を行なってください．既にF-BASICで使用中のディスクなら，ファイルはそのまま残っており，すぐに使えます．

FM-7のパワーオン，またはRESETボタンを押すと，IODOS9がロードされ実行されます．

ブート時のディップ・スイッチは，F-BASICモード，DOSモードのいずれでもかまいません．ブート時には，メモリをクリアせずに，BIOSをROMから裏RAMにコピー，画面を80×25に設定，先行入力を可能でコマンド待ちになります．ただし，BASICテキストはNEWされます．

＊1：F-BASICのフォーマッティング・ルーチン(SYSDSK）を用いてください．URADOSのフォーマッティング・ルーチンを用いると，IODOS9でもロード，セーブが一段と早くなります．

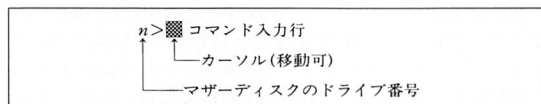
スタートアップ・プログラム

ブート時に，任意のマシン語ファイルをロードし，実行させることができます．非常駐コマンド**STARTUP**を実行させると，スタート・アップするマシン語ファイルのファイル名を指定できます．

この機能は，ディップ・スイッチがDOSモードになっているときのみ，動作します．スタート・アップしたプログラムからIODOSに戻るには，リターン命令(**RTS**など)を使ってください．

コマンドの入力

IODOS9が">"を表示しているときは，コマンドが入力できます．

```
n>▩ コマンド入力行
  └─カーソル(移動可)
└───マザーディスクのドライブ番号
```

マザーディスクとは，常駐コマンドでドライブ番号を省略したときに有効になるドライブ番号です．

コマンドには，**常駐**（メモリ上に存在している）コマンド，**非常駐**（ディスク上に存在している）コマンド，それに**制御**コマンドがあります．

非常駐コマンド名の前に，"ドライブ番号:" を付けると，その非常駐コマンドの入っているドライブが指定できます．

画面に表示されているコマンドは，カーソル移動して，一部を変更すると，再び実行されます（たとえば，スペースを挿入して⏎する）．

制御コマンド

❶マザーディスクの変更

数字 "0"，または "1" を指定すると，マザーディスクがその数値のドライブに変更されます．

例．0>1 RETURN とすると，
1>▩ となる．

❷プリンタ・スイッチ

記号 "+" を指定すると，プリンタ出力可能にします．この状態での（IODOS9）の画面への出力は，すべて同時にプリンタにも出力され，再度，"+" を指定すると，プリンタ出力をしません．

例．0>+ RETURN とすると，"Printer on" のメッセージが出力され，プリンタ出力モードになったことを示します．

❸コマンド入力ルーチンを元に戻す．

16進数値$1Aを入力すると，コマンド入力ルーチンを，IODOS9のキーボードからの一行入力ルーチンに戻します．

常駐コマンド

6種の常駐コマンドがあります．このコマンドは，大文字，小文字のいずれで指定しても動作可能です．

以下，< >は省略可能を，{ }はいずれかを選択することを意味し，アドレスの16進数は，"&H" をつけても，つけなくても構いません．**dr**はドライブ番号で，0または1を意味します．␣は空白1個を意味します．

❶LOAD

機能：マシン語または，BASICプログラム・ファイルを，ディスクからメモリにロードする．

形式：**LOAD<M>** "<**dr**:>ファイル名<" <, <オフセット><, **R**>>>

説明

オフセット：マシン語ファイルについてのみ有効．本来ロードするアドレスに，加える，または引く数値を16進数で指定する．引くときは，16進数の前に "—"(マイナス)をつける．

R：マシン語ファイルについてのみ有効．ロード後，直ちに実行する．

注) ASCII形式のファイルは，ロードできない．

❷SAVE

機能：マシン語，またはBASICプログラム・ファイルを，メモリからディスクへセーブする．

形式：**SAVE<M>** "<**dr**:> ファイル名<" <, 開始アドレス, 終了アドレス <, 実行アドレス>>>

説明

BASICプログラムの場合は，ファイル名のみでよいが，マシン語ファイルでは，先頭アドレスと終了アドレスを指

定する．実行アドレスは実行開始アドレスで，省略すると先頭アドレスとされる．

❸FILES

機能：ディスクの中のファイルを，リスト・アップする．
形式：**FILE**<S><"<**dr**：><ファイル名><">>
説明：
ファイル名の指定がないときは，すべてのファイルをリスト・アップする．ファイル名を指定したときは，"＊"がくるまで一致した*fn*または*ft*をもつファイルをリスト・アップする．
ファイル属性は，ファイル名の右に表示される．

A：ASCII形式
B：BASICプログラム形式
M：マシン語形式

> 例 FILES "BA＊␣COM"
> *fn*が"BA"で始まるファイルで，*ft*が"COM"のものを表示する．

❹KILL

機能：ファイルを削除する．
形式：**KILL** "<**dr**：> ファイル名 <">
説明："Are you sure (Y or N)?" のメッセージが出されるので，[Y] (Yes) または [N] (No) を入力し，[RETURN] キーを押す．

❺NAME

機能：ファイル名を変更する．
形式：**NAME** "<**dr**：> 旧ファイル名 "AS" 新ファイル名"
説明：新ファイル名と同じファイル名が既にあると，エラーになる．新ファイル名は，旧ファイル名と同じドライブ上につくられる．

❻EXEC

機能：あるアドレスから実行する．
形式：**EXEC**␣アドレス
説明：メモリ上にあるマシン語プログラムを実行する．アドレスは16進数で入れる．**EXEC**とアドレスの間には，1つ以上の空白が必要です．
マシン語プログラムの中では，レジスタ類を自由に使って良い．IODOS 9 に戻るには，リターン命令（**RTS**など）を使う．

非常駐コマンド

非常駐コマンドは，ディスクからコマンド領域にロードされて実行されるコマンドです．ファイル名を変更するだけで，コマンドとして利用可能です．

BIOSを使うリロケータブルなプログラムなら，非常駐コマンドとなります．この例として，笠作貴弥氏の"6809デバッガ"（FM-8活研）や，馬場二郎氏の"エディタ・アセンブラVer2.0"（FM-7/8活研）など，便利に使えるようです．

この他に，IODOS 9 のランタイム・ルーチンや，入出力ルーチンを使うリロケータブルなプログラムも，非常駐コマンドとなります．これらの例として，いくつかの非常駐コマンドとその作り方を以下に示します．

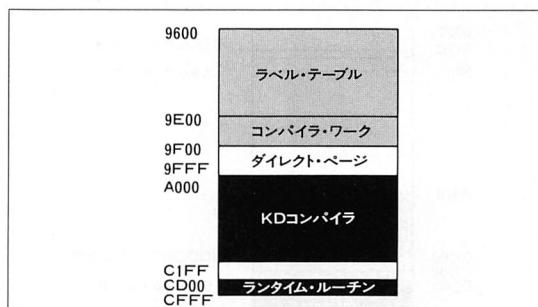



図2　KDコンパイラ・メモリ・マップ

❶F-BASIC

機能：IODOS 9 から，F-BASICに制御を移す．
形式：FBASIC
説明：F-BASICに制御を移す．このF-BASICは，ROMモードで，テキスト・エリアは，$900～$7FFFまでである．
このコマンドは，F-BASICになっても，0～$8FF以外はメモリ・クリアをしない**NEW**の状態に，テキストがある場合はその状態を保持する．
IODOS 9 に戻るときは，F-BASICに拡張してあるコマンドの "**BYE**" を使う．直接またはプログラム中から，**BYE**を実行させると，IODOS 9 に戻る．
作り方：アセンブル・リストを，**リスト2**に示す．マシン語の部分を打ち込んで，ファイル名を"**FBASIC␣COM␣R**"としてセーブする．
注：F-BASICにおいて，$FC00～$FC2Fを壊すと，**BYE**コマンドでIODOS 9 に戻れなくなる．
もし，何らかの理由でIODOS 9 に戻れなくなったときは，RAMモードにして，$D000にジャンプすればIODOS9に戻る（たとえば，RAM上に7FFD0F7ED000を書き，この先頭アドレスにジャンプする）．

❷KD（IODOSバージョンKコンパイラ）

機能：IODOS 9 のランタイム・ルーチン（$CD00～$CFFF）を利用して，メモリ上の，Kコンパイラのソース・プログラムをコンパイルする．
形式：**KD**
説明：BASICテキストの形で，メモリ上にあるプログラムをコンパイルする．オブジェクトは，$5000番地から作成され，IODOS 9 のランタイム・ルーチンを使って実行する．
作り方：拡張Kコンパイラ（バージョン3.6）を，**リスト3**に従って変更する．変更したら，ディスクにセーブする．IODOS 9 をブートして，ファイル名を，"**KD␣COM␣R**"に変更する．コンパイル時のメモリ・マップを**図2**に示す．

❸TYPE

機能：ディスク上のファイルについての情報を出力する．
形式：**TYPE** "<**dr**:> ファイル名<">
説明：次のものを表示する．
（i）ファイル名
（ii）DATE（もし指定されていれば）
（iii）ディレクトリの位置（トラック，セクタ）
（iv）ファイルの先頭位置（トラック，セクタ）
（v）使用しているクラスタ番号を，使用している順に

図3　メモリ・マップ

図4　FCB (File Control Block)

0	▼file name (fn)*1	
2		
4		
6	*1 *fn* 以外は空白($20)を入れる.	
8	▼file type (ft)*2　*2 *ft* 以外はヌル(0)を入れる.	
10		▽ファイル・タイプ
12	▽ASCIIフラグ	▽ランダム・アクセス・フラグ
14	先頭クラスタ・アドレス	▼file mode (fm)*3
16	ワーク	エラーフラグ
18	▼バッファ・アドレス*4　*4 バッファ・サイズは256バイト.	
20	トラック(0～39)	セクタ(1～16)
22	サイド(0～1)	▼ドライブ番号(0～1)
24	アクセス中のクラスタ・カウンタ	アクセス中のクラスタ・アドレス
26	ディレクトリのトラック	ディレクトリのセクタ
28	ディレクトリのサイド	ディレクトリ上のアドレス
30	▼open mode (0……read / 1……write)	open完了フラグ
32	コメント*5	
34		
36		
38		
40		
42		
44		
46		

*3 *fn* を指定しないときはヌル(0)にする.
*5 read open時：ディレクトリの16～32バイトに書かれている内容が入る.
write close時：ディレクトリの16～32バイトにコメントを書きこむ.

①▼：open時に必ず指定するもの.
②▽：write close時(open時でもよい)に必ず指定するもの.
③その他：open時，またはread/write時にセットされる.
④FCBは，ファイルをCLOSEするまで，壊さないこと.

16進数で表示.

(vi)ファイル属性

(vii)ファイル内容：マシン語，BASICファイルでは，先頭アドレス，終了アドレス，実行アドレスを表示する．ASCIIファイルでは，書かれている内容を表示する．

作り方：リスト4のプログラムを，F-BASICのエディタを使って作る．その後，KDコンパイラでコンパイルし，オブジェクトを〝TYPE␣COM␣R〟の名前でSAVEする．

❹WIDTH

機能：画面の一行の文字数を変更する．

形式：WIDTH〈␣*n*〉

説明：*n*が40のときは，一行に40文字表示する．*n*が40以外では，一行に80文字表示する．行数は，いずれのときも25行の設定になっている．

作り方：リスト5のプログラムを，F-BASICのエディタを使って作る．あとは，KDコンパイラでコンパイルし，オブジェクトを〝WIDTH␣COM␣R〟のファイル名でSAVEする．

❺DATE

機能：ディレクトリに書きこむメッセージを指定したり，指定されているメッセージを表示する．

形式：DATE〈␣16文字内のメッセージ〉

説明：文字列として日付などを指定すると，それ以降SAVEコマンドでファイルを作るたびに，そのディレクトリに指定したメッセージを書き込む．使える文字は，キーボードから入る文字ならなんでも良い．

書き込んだメッセージは，〝TYPE〟コマンドで見ることができる．

パラメータの指定がないときは，現在設定されているメッセージを表示する．

作り方：リスト6のプログラムを，F-BASICのエディタを使って作り，KDコンパイラでコンパイル後，オブジェクトを〝DATE␣COM␣R〟のファイル名でSAVEする．

❻HELP

機能：IODOS9のコマンドの使い方を表示する．

形式：HELP

作り方：リスト7のプログラムを，F-BASICのエディタを使って作成する．次に，KDコンパイラによりコンパイルする．コンパイルできたら，オブジェクトを〝HELP␣COM␣R〟のファイル名でセーブする．

❼STARTUP

機能：ディスクのボリューム名と，ブート時に実行するプログラム名を登録する．

形式：STARTUP

説明：2文字のボリューム名と，ブート時に実行するプログラム名（8文字内の*fn*で*ft*, *fm*は無指定とする）を登録する．

ボリューム名を指定しておくと，OPEN，CLOSE時に同一ボリュームかをチェックする．

ブート時に実行するプログラム（スタートアップ・プログラム）は，ここで指定したファイル名を持つ，実行可能なマシン語ファイルで，ブートされるたびに最初に実行される．ファイルがないと，〝File Not Found〟のメッセージが出力され，システムに戻る．

スタート・アップは，DOSモードのときのみ動作するので，ディップ・スイッチを切り変えておく．

使い方：リスト8のプログラムを，F-BASICのエディタを使って作る．あとは，KDコンパイラでコンパイルし，ファイル名を〝STARTUP␣COM␣R〟としてセーブする．

終わりに

IODOS9，いかがでしたか．いままで中央にデンと居座っていた，F-BASICが一つのモードになってしまいました．

IODOS9は，BASICインタープリタより，コンパイラやマシン語中心に利用する人のためのDOSです．しかも，F-BASICと互換性もあります．

BIOSを利用するリロケータブルなプログラムなら，すぐにコマンドになります．あなたの工夫次第で，IODOS9は無限に発展する可能性を秘めています．仕事や，ゲームのプログラム開発に活用し，役立てていただければ幸いです．

表1　入出力サブルーチン

両面，キーボード関係

名称	アドレス	機能
OUTEE	CD00	Aレジスタの値を，画面またはプリンタに出力する．$CF38が$20なら画面に出力し，$8Dなら画面とプリンタに出力する．レジスタは保存される．
INEE	CD03	キーボードから1文字Aレジスタに入力する．キー入力なしは，0(ヌル)となる．エコーバックはない．Aレジスタ以外保存される．
GETLINE	CD06	Xレジスタで示されるバッファに，キーより一行(⏎が押されるまで)を入力する．一行の終わりには$0Dが入る．レジスタは保存される．CTRL+C でキー入力を中止する．

フロッピーディスク関係

Xレジスタに，FCB (File Control Block) のアドレスをセットして，サブルーチン・コールする．レジスタ類はすべて保存される．サブルーチンから戻ったら，必ずエラーフラグを確かめる．

名称	アドレス	機能
FOPEN	D018	ファイルのOPENを行なう．
FCLOSE	D01B	ファイルのCLOSEを行なう．OPENしたファイルは，必ずCLOSEする．
FREAD	D01E	OPEN済のファイルを1セクタずつ読む．
FWRITE	D021	OPEN済のファイルに1セクタずつ書く．

表2　エラーメッセージ

①コマンド解釈からのメッセージ

FORMAT ER	コマンドのフォーマットがおかしい．
ADDRESS ER	アドレスの指定がおかしい
DEVICE ER	デバイス番号がおかしい．

②ディスク管理からのメッセージ

File Not Found	ファイル名が見つからない．
File Already Exist	ファイル名がすでに存在する．
Disk Full	ディスクがいっぱいである．
Bad File Mode	ファイル形式が違う．
Protected File	プロテクトされたファイルである．
Drive Not Ready	ドライブがReady状態にない．
Disk Write Protected	ディスクが書き込み保護されている．
FAT ER	ファットの中味がおかしい．
OPEN ER CODE=※	ファイル・オープン時にエラーが生じた．
CLOSE ER CODE=※	ファイル・クローズ時にエラーが生じた．
READ ER CODE=※	ファイル・リード時にエラーが生じた．
WRITE ER CODE=※	ファイル・ライト時にエラーが生じた．

注）※は，エラーコードで，表3を参照のこと．

表3　エラーコード(FCBのエラーフラグにセットされる)

コード	内容
0	normal end
10	drive not ready
11	disk write protected
12	hard error
13	CRC error
14	DD mark detected
15	time over
16	file not found
17	bad file mode
18	file already exist
19	no disk space
20	directory over
21	end of file
22	file not opened
23	file is protected

表4　主なエントリ

アドレス	内容
D000	IODOS9ホット・スタート
D010	1行入力ルーチン・アドレス．普通は$CD06になっている．EOF($1A)がくると，$CD06にリセットされる．
D013	フロッピーディスクRESTOREルーチン・アドレス
D016	フロッピーディスクREAD/WRITEルーチン・アドレス
D024	主アクセス・ドライブ番号が入っている．
D02B	Breakエントリ．
EC00	FAT用エリア．
EE00	FCB用エリア．
EE30	FAT用FCB用エリア．
EE50	DATE用エリア．
EE60	パラメータ用エリア．
EE80	ワーク用エリア．

リスト1（その1）IODOS 9 作成プログラム　ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1B00 1A 50 86 FD 1F 8B 20 0B 30 8D 00 31 86 81 97 03 :51
1B10 BD FE 02 30 8D 00 2E BD FE 08 25 EC A6 08 4A 27 :9B
1B20 18 A7 08 EC 02 C3 01 00 ED 02 A6 05 4C 81 11 26 :17
1B30 04 86 01 A7 06 A7 05 20 DA 4F 1F 8B 39 08 00 00 :18
1B40 00 00 00 00 00 09 00 1C 00 00 01 00 00 20 00 00 :46
1B50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1B60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1B70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1B80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1B90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1BA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1BB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1BC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1BD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1BE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1BF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: F3 7B 91 C0 B4 FE 54 04 F5 E6 EB AD B1 32 F2 50 :61
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1C00 32 8C FD 20 09 00 00 00 00 00 00 00 00 00 7F FD :60
1C10 0F BF FF F6 1A 50 86 FD 1F 8B 96 04 85 02 27 37 :D9
1C20 20 0F 30 8D 00 A5 86 81 5C 5C ED 84 97 03 BD FE :16
1C30 02 30 8D 00 9E BD FE 08 25 E8 A6 08 4A 27 18 A7 :0B
1C40 08 EC 02 C3 01 00 ED 02 A6 05 4C 81 11 26 04 86 :E2
1C50 01 A7 06 A7 05 20 DA B6 FD 05 2B FB 1A 40 86 80 :92
1C60 B7 FD 05 B6 FD 05 2A FB CC 0C 12 FD FC 82 7F FD :77
1C70 05 7D FD 0F BE FB FA A6 84 7F FD 0F A7 80 7D FD :97
1C80 0F 8C FC 00 26 F1 7F FD 0F 8E FB FC CE D0 00 EF :4B
1C90 81 EF 81 8E 03 02 10 8E 1B 67 8D 1F BF FC 82 10 :9D
1CA0 BF FC 84 7F FD 05 4F 1F 8B 8E 09 00 9F 33 4F 5F :D0
1CB0 ED 81 9F 35 9F 3B 9F 3D 7E D0 00 B6 FD 05 2B FB :24
1CC0 86 80 B7 FD 05 B6 FD 05 2A FB 39 08 00 00 00 00 :DD
1CD0 00 00 00 0A 00 CD 00 00 02 00 00 1F 00 00 00 00 :F8
1CE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1CF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: EA 0F 1A 1B 4C 88 6F CB F2 B2 79 10 5D 98 FD 32 :8D
```

リスト1（その2）

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1D00 16 02 35 16 02 79 17 02 99 20 0C 35 10 20 02 8D :B0
1D10 EF A6 80 26 FA 6E 84 86 0D 8D E5 86 0A 20 E1 36 :F3
1D20 06 A6 43 3D 34 06 EC 41 3D EB E4 E7 E4 A6 C4 E6 :BA
1D30 43 3D EB E0 A6 E0 1E 89 33 44 39 8E 00 00 34 10 :FA
1D40 34 10 34 10 4D 2A 04 8D 40 63 61 6C 60 2B 40 58 :23
```

```
1D50 49 2A F8 44 56 ED 64 EC C1 2A 04 8D 2C 63 61 6A :18
1D60 60 2B 1E 58 49 2A F8 44 56 A3 64 20 06 58 49 24 :F8
1D70 F8 E3 64 1C FE 2B 02 1A 01 69 63 69 62 6A 60 2A :2C
1D80 EC EC 62 6D 61 32 66 2A 05 43 53 C3 00 01 39 32 :94
1D90 66 CC 7F FF 39 4D 2B F1 39 A3 C1 2F 2C CC 00 00 :16
1DA0 39 A3 C1 2E 24 CC 00 00 39 A3 C1 2D 1C CC 00 00 :6D
```

リスト1(その2)

```
1DB0 39 A3 C1 2C 14 CC 00 00 39 A3 C1 27 0C CC 00 00 :45
1DC0 39 A3 C1 26 04 CC 00 00 39 CC 00 01 39 83 00 00 :55
1DD0 27 F7 CC 00 00 39 17 FF 2A 1F 89 4F 39 8D 15 16 :4B
1DE0 FF A7 8E 04 3D 17 FF 1E A6 80 81 24 27 2B 81 2D :74
1DF0 27 EB 30 1F CC 00 00 34 06 A6 84 82 30 2B 18 81 :07
-----------------------------------------------------------
Sum: 6D FD 3F 30 9F 6C AE 95 2D B2 5E EE 0F 01 0C BF :2D

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1E00 09 22 14 A7 84 EC 60 58 49 58 49 E3 60 58 49 EB :C7
1E10 80 89 00 ED 60 20 E2 35 86 CC 00 00 34 06 A6 84 :43
1E20 82 30 2B F3 81 09 23 06 82 07 81 0F 22 E9 A7 84 :D2
1E30 EC 60 58 49 58 49 58 49 58 49 EB 80 ED 60 20 DE :86
1E40 8E 00 00 36 10 36 10 36 10 36 10 4D 2A 05 63 40 :C5
1E50 17 00 D7 58 49 ED 46 8E 00 0F C6 05 1C FE A6 C5 :AF
1E60 49 81 09 22 04 1C FE 20 04 82 0A 1A 01 A7 C5 5A :A4
1E70 26 EC 68 47 69 46 24 02 6C 45 30 1F 26 DC C6 01 :5F
1E80 A6 C5 26 05 5C C1 05 26 F7 39 8D B4 34 04 EB 49 :BB
1E90 C2 06 23 02 8D 20 35 04 6D 40 2A 05 86 2D 17 FE :77
1EA0 5F A6 C5 8B 30 17 FE 58 5C C1 06 26 F4 33 4A 39 :E5
1EB0 33 5E 8D 8C 20 E2 86 20 17 FE 45 5A 26 FA 39 34 :93
1EC0 04 8D 04 35 04 1F 98 1F 89 44 44 44 44 8D 04 1F :ED
1ED0 98 84 0F 8B 30 81 39 23 02 8B 07 16 FE 22 1F 98 :44
1EE0 20 F9 33 5C A6 45 E6 47 3D ED 42 EC 45 3D EB 42 :C7
```

```
1EF0 89 00 ED 41 A6 44 E6 47 3D E3 41 ED 41 A6 44 E6 :2D
-----------------------------------------------------------
Sum: 4A 81 AD 42 3C E6 90 34 05 57 95 69 AC 1D 21 C4 :A8

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1F00 46 3D EB 41 89 00 ED C4 39 34 06 EC 8C 1A 8E 3D :B9
1F10 09 36 16 8D CD EC 42 33 48 C3 00 03 ED 8C 09 35 :D5
1F20 10 36 16 8D BD 33 48 39 1C 2C 6A 49 16 FE 5A 20 :E3
1F30 7C CF 34 3B 8C 00 00 00 20 02 20 13 34 31 8D 26 :B3
1F40 10 8E FC 82 8E 03 01 AF A1 A7 A4 8D 2E 35 B1 34 :1E
1F50 05 F6 FD 02 54 25 FA B7 FD 01 C6 00 F7 FD 00 CA :A6
1F60 40 F7 FD 00 35 85 34 03 B6 FD 05 2B FB 1A 40 86 :E3
1F70 80 B7 FD 05 B6 FD 05 2A FB 35 83 7F FD 05 39 34 :BC
1F80 05 C6 12 8D 5D 8D DF CC 29 00 FD FC 82 8D EC 8D :A9
1F90 D5 8D 02 35 85 C6 80 B6 FC 83 FA FC 80 F7 FC 80 :82
1FA0 20 D9 34 35 C6 13 8D 3A 8D BC CC 04 02 FD FC 82 :98
1FB0 CC 11 07 FD FC 84 8D C3 8D AC F6 FC 83 26 2C F6 :A7
1FC0 FC 84 27 12 5A 5A 5A 27 0D 10 8E FC 88 A6 A0 27 :8A
1FD0 05 A7 80 5A 26 F7 C6 0D E7 84 8D B9 C6 12 8D 02 :8E
1FE0 35 B5 8D 82 86 0C FD FC 82 20 90 4F 1F 8B 8D A5 :E1
1FF0 7E D0 00 81 FF 27 06 7D FD 0F 7E 8F D0 7E 94 8D :00
-----------------------------------------------------------
Sum: 2A 97 C1 82 15 37 47 EF BE AD 64 0D A4 8E 06 50 :EA
```

リスト1(その3)

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2000 34 7F C6 CC 1F 9B CE FC 80 10 DF FE 16 00 48 7E :12
2010 CD 06 7E FE 02 7E FE 08 16 12 3D 16 12 48 16 12 :D2
2020 53 16 12 5E 00 17 12 68 16 00 32 C6 40 F7 FD 05 :B1
2030 B6 FD 04 85 02 27 04 7F FD 05 3B 7F FD 05 F6 FD :99
2040 04 C5 02 27 F9 17 12 48 BD CD 17 BD CD 0B 42 72 :46
2050 65 61 6B 00 16 00 C9 CC 00 00 F7 D0 24 8E D0 2B :50
2060 BF FF F6 CC 00 00 DD 00 CC 00 00 DD 02 CC EE 60 :22
2070 DD 04 CC EE 00 DD 06 CC EE 80 DD 08 CC EE 50 DD :84
2080 0A 4F F6 D0 24 DD 00 CC EF 00 DD 0E 0F 01 DC 00 :BE
2090 83 00 7F 10 2E 00 14 DC 00 58 49 9E 06 30 8B 36 :66
20A0 10 CC 00 00 ED D1 0C 01 16 FF E3 CC 03 05 DD 10 :60
20B0 0F 01 DC 00 9E 10 E6 8B 4F 83 00 20 10 2D 00 16 :50
20C0 DC 00 9E 06 30 8B 36 10 DC 00 9E 10 E6 8B 4F E7 :B2
20D0 D1 0C 01 16 FF DC DC 00 83 00 08 10 26 00 18 CC :50
20E0 00 00 9E 10 E7 84 CC 00 00 DD 12 DC 0C DD 14 CC :79
20F0 00 01 DD 16 16 07 E9 BD CD 17 BD CD 0B 49 4F 44 :0C
-----------------------------------------------------------
Sum: 68 EA F4 B0 3B FB 79 CC A0 42 F2 2C 6F AB AF 8B :C5

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2100 4F 53 39 20 56 65 72 20 31 2E 30 20 28 43 29 57 :E2
2110 6F 6E 64 65 72 20 53 6F 66 74 20 27 38 34 2E 00 :B5
2120 CC D0 10 DD 18 CC CD 06 9E 18 ED 84 10 CE 08 FF :4C
2130 BD CD 17 DC 0C BD CE B0 BD CD 0B 3E 00 CC EF 00 :52
2140 DD 0E CC EE 00 DD 06 CC 00 00 DD 12 CC 00 00 DD :EC
2150 1A DC 0C DD 14 DC 0C F7 D0 24 1C 00 17 06 38 17 :4E
2160 06 54 17 06 3E 1A 50 30 8C 02 20 05 4C 4F 41 44 :22
2170 00 34 10 17 06 57 32 62 10 27 00 03 16 06 BF 30 :91
2180 8C 02 20 05 53 41 56 45 00 34 10 17 06 3F 32 62 :16
2190 10 27 00 03 16 09 36 30 8C 02 20 05 46 49 4C 45 :92
21A0 00 34 10 17 06 27 32 62 10 27 00 05 1C 00 16 0B :95
21B0 24 30 8C 02 20 05 4B 49 4C 4C 00 34 10 17 06 0D :A1
21C0 32 62 10 27 00 03 16 0E F1 30 8C 02 20 05 4E 41 :55
21D0 4D 45 00 34 10 17 05 F5 32 62 10 27 00 03 16 0D :D8
21E0 E1 30 8C 02 20 05 45 58 45 43 00 34 10 17 05 DD :26
21F0 32 62 10 27 00 05 1C 00 16 0F 20 9E 10 4F E6 84 :98
-----------------------------------------------------------
Sum: 96 96 2B CB 03 D2 79 15 C4 61 4D 73 6D 79 6F 2C :EB

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2200 83 00 0D 10 26 00 03 16 FF 29 9E 10 4F E6 84 83 :F1
2210 00 1A 10 26 00 03 16 FF 07 9E 10 4F E6 01 83 00 :D6
2220 0D 10 26 00 7A 9E 10 4F E6 84 83 00 2B 10 26 00 :08
2230 3D 4F F6 CF 38 DD 18 BD CD 0B 50 72 69 6E 74 65 :85
2240 72 20 00 DC 18 83 00 20 10 26 00 0F BD CD 0B 6F :72
2250 6E 00 CC 00 8D F7 CF 38 16 00 0D BD CD 0B 6F 66 :52
2260 66 00 CC 00 20 F7 CF 38 16 FE C1 16 00 31 9E 10 :1A
2270 4F E6 84 36 06 CC 00 30 BD CD B9 36 06 9E 10 4F :6D
2280 E6 84 36 06 CC 00 31 BD CD B9 AA C0 EA C0 10 27 :31
2290 00 0D 9E 10 4F E6 84 83 00 30 DD 0C 16 FE 94 DC :94
22A0 06 34 06 17 02 D3 32 62 CC 04 3D DD 10 9E 10 4F :B7
22B0 E6 84 83 00 30 10 2D 00 0A DC 10 C3 00 01 DD 10 :01
22C0 16 FF EA 17 04 F0 0F 03 9E 10 4F E6 84 36 06 CC :8B
22D0 00 0D BD CD C1 36 06 DC 02 36 06 CC 00 1F BD CD :23
22E0 A1 A4 C0 E4 C0 10 27 00 1B DC 02 9E 04 30 8B 36 :6C
22F0 10 9E 10 4F E6 84 E7 D1 0C 03 DC 10 C3 00 01 DD :CB
-----------------------------------------------------------
Sum: FB 16 29 5B 5B 3E 16 33 1C 35 0F B5 B4 EE A9 2A :01

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2300 10 16 FF C4 DC 02 9E 04 30 8B 36 10 CC 00 00 E7 :1D
2310 D1 0F 03 DC 02 83 00 07 10 2E 00 1D DC 02 9E 06 :28
2320 30 8B 36 10 DC 02 9E 06 E6 8B 4F 34 06 17 04 E1 :79
2330 32 62 E7 D1 0C 03 16 FF DA CC 43 4F 9E 06 ED 08 :41
2340 CC 00 4D 9E 06 E7 0A CC 00 52 9E 06 E7 0F CC 00 :32
2350 01 DD 1A CC 00 01 DD 16 16 05 85 BD CD 0B 46 4F :82
2360 52 4D 41 54 20 00 16 00 3E DC 14 36 06 CC 00 00 :A0
2370 BD CD B9 36 06 DC 14 36 06 CC 00 01 BD CD B9 AA :65
2380 C0 EA C0 10 27 00 01 39 BD CD 17 BD CD 0B 44 45 :9A
2390 56 49 43 45 20 00 0C BE 16 00 0C BD CD 0B 41 44 :4D
23A0 44 52 45 53 53 20 00 BD CD 0B 45 52 00 16 FD 7C :5C
```

```
23B0 9E 06 4F E6 88 11 83 00 10 10 26 00 15 BD CD 0B :E5
23C0 46 69 6C 65 20 4E 6F 74 20 46 6F 75 6E 64 00 16 :03
23D0 FD 5A 9E 06 4F E6 88 11 83 00 12 10 26 00 19 BD :6A
23E0 CD 0B 46 69 6C 65 20 41 6C 72 65 61 64 79 20 45 :9F
23F0 78 69 73 74 00 16 FD 34 9E 06 4F E6 88 11 83 00 :04
-----------------------------------------------------------
Sum: 9F CB DA 4B EF 2E 07 D6 B7 B5 C2 42 F2 A9 65 F7 :F0

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2400 13 10 26 00 10 BD CD 0B 44 69 73 6B 20 46 75 6C :C0
2410 6C 00 16 FD 17 9E 06 4F E6 88 11 83 00 14 10 26 :D5
2420 00 15 BD CD 0B 44 69 72 65 63 74 6F 72 79 20 46 :C5
2430 75 6C 6C 00 16 FC F5 9E 06 4F E6 88 11 83 00 11 :5A
2440 10 26 00 14 BD CD 0B 42 61 64 20 46 69 6C 65 20 :A6
2450 4D 6F 64 65 00 16 FC D4 9E 06 4F E6 88 11 83 00 :60
2460 17 10 26 00 15 BD CD 0B 50 72 6F 74 65 63 74 65 :3D
2470 64 20 46 69 6C 65 00 16 FC B2 9E 06 4F E6 88 11 :3A
2480 83 00 0A 10 26 00 16 BD CD 0B 44 72 69 76 65 20 :88
2490 4E 6F 74 20 52 65 61 64 79 00 16 FC 8F 9E 06 4F :DA
24A0 E6 88 11 83 00 0B 10 26 00 1B BD CD 0B 44 69 73 :13
24B0 6B 20 57 72 69 74 65 20 50 72 6F 74 65 63 74 65 :FC
24C0 64 00 16 FC 67 BD CD 0B 4F 50 45 4E 20 45 52 20 :7B
24D0 43 4F 44 45 3D 00 9E 06 4F E6 88 11 BD CE B0 16 :1B
24E0 FC 4A BD CD 0B 43 4C 4F 53 45 20 45 52 20 43 4F :BA
24F0 44 45 00 9E 06 4F E6 88 11 BD CE B0 16 FC 2D BD :32
-----------------------------------------------------------
Sum: D5 4B 32 7D 1C D3 8E F0 78 01 9B 8E F5 06 43 08 :24

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2500 CD 0B 52 45 41 44 20 45 52 20 43 4F 44 45 3D 00 :23
2510 9E 06 4F E6 88 11 BD CE B0 16 FC 10 BD CD 0B 57 :BB
2520 52 49 54 45 20 45 52 20 43 4F 44 45 3D 00 9E 06 :07
2530 4F E6 88 11 BD CE B0 16 FB F2 BD CD 0B 46 41 54 :7C
2540 20 45 52 00 16 FB 10 17 02 6C 9E 10 4F E6 84 83 :47
2550 00 22 10 27 00 1C 9E 10 4F E6 84 83 00 0D 10 26 :A2
2560 00 06 CC 00 01 ED 62 39 DC 10 C3 00 01 DD 10 16 :0E
2570 FF DB DC 10 C3 00 01 DD 10 17 02 3A EC 62 DD 1C :0E
2580 0F 01 DC 00 83 00 07 10 2E 00 12 DC 00 9E 1C 30 :8C
2590 8B 36 10 CC 00 20 E7 D1 0C 01 16 FF E5 CC 00 00 :48
25A0 9E 1C ED 08 CC 00 00 9E 1C E7 0A CC 00 00 9E 1C :AC
25B0 E7 0F 0F 01 0F 03 9E 10 4F E6 84 83 00 22 10 27 :5B
25C0 00 B6 9E 10 4F E6 84 83 00 20 10 26 00 07 0C 03 :0C
25D0 0F 01 16 00 99 9E 10 4F E6 84 83 00 3A 10 26 00 :19
25E0 16 9E 1C 4F E6 84 83 00 30 DD 14 0F 01 CC 00 20 :29
25F0 9E 1C E7 84 16 00 77 9E 10 4F E6 84 83 00 0D 10 :B9
-----------------------------------------------------------
Sum: 0D 58 26 70 C2 97 0A 85 48 8E 6A 21 28 F9 B1 32 :48

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2600 26 00 03 16 00 7C DC 02 83 00 00 10 26 00 1B DC :49
2610 00 83 00 07 10 2E 00 0F DC 00 9E 1C 30 8B 36 10 :6E
2620 9E 10 4F E6 84 E7 D1 16 00 42 DC 02 83 00 01 10 :E9
2630 26 00 1E DC 00 83 00 02 10 2E 00 12 DC 00 C3 00 :94
2640 08 9E 1C 30 8B 36 10 9E 10 4F E6 84 E7 D1 16 00 :F8
2650 1B DC 02 83 00 02 10 26 00 12 DC 00 83 00 00 10 :35
2660 26 00 09 9E 10 4F E6 84 9E 1C E7 0F 0C 01 DC 10 :3F
2670 C3 00 01 DD 10 16 FF 3E DC 10 C3 00 01 DD 10 17 :B8
2680 01 34 DC 02 36 06 CC 00 00 BD CD B9 36 06 DC 00 :76
2690 36 06 CC 00 00 BD CD B9 A4 C0 E4 C0 10 27 00 08 :92
26A0 CC 00 01 ED 62 16 00 05 CC 00 00 ED 62 39 EC 62 :D9
26B0 34 06 17 01 5C 32 62 ED 62 EC 62 36 06 CC 00 30 :17
26C0 BD CD 99 36 06 EC 62 36 06 CC 00 39 BD CD B1 A4 :CD
26D0 C0 E4 C0 10 27 00 0A EC 62 83 00 30 DD 1E 16 00 :B7
26E0 05 CC FF FF DD 1E EC 62 36 06 CC 00 41 BD CD 99 :84
26F0 36 06 EC 62 36 06 CC 00 46 BD CD B1 A4 C0 E4 C0 :1B
-----------------------------------------------------------
Sum: E5 D0 9C A4 73 CC D1 DE AF 78 92 89 59 D4 57 CA :73

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2700 10 27 00 0B EC 62 83 00 41 C3 00 0A DD 1E 39 DC :31
2710 1E ED 62 39 17 00 9F 9E 10 4F E6 84 36 06 CC 00 :CB
```

リスト1（その3）

```
2720 26 BD CD B9 36 06 9E 10 4F E6 01 34 06 17 00 E1 :BB
2730 32 62 36 06 CC 00 48 BD CD B9 A4 C0 E4 C0 10 27 :66
2740 00 07 DC 10 C3 00 02 DD 10 CC 00 00 DD 20 0F 01 :7E
2750 9E 10 4F E6 84 34 06 17 FF 54 32 62 36 06 CC 00 :A7
2760 00 BD CD 99 36 06 DC 00 36 06 CC 00 03 BD CD B1 :81
2770 A4 C0 E4 C0 10 27 00 1A DC 10 C3 00 01 DD 10 DC :D2
2780 20 58 49 58 49 58 49 58 49 D3 1E DD 20 0C 01 16 :B5
2790 FF BE DC 20 DD 20 39 8E 04 3D 17 FB 72 CC 04 3D :4C
27A0 DD 10 39 8E 03 02 CC 11 07 BD CF 66 BF FC 82 FD :C9
27B0 FC 84 7F FD 05 39 9E 10 4F E6 84 83 00 20 10 26 :7A
27C0 00 0A DC 10 C3 00 01 DD 10 16 FF EA 39 EC 62 DD :0A
27D0 18 0F 01 DC 00 9E 18 E6 8B 4F 83 00 00 10 27 00 :34
27E0 29 DC 00 9E 10 E6 8B 4F 34 06 17 00 24 32 62 36 :B2
27F0 06 DC 00 9E 18 E6 8B 4F BD CD C1 10 27 00 06 CC :AC
--------------------------------------------------------
Sum: 07 42 FB 7D AB E6 07 E1 BD D2 2E 9C E9 DD 55 C7 :75

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2800 00 00 DD 18 39 0C 01 16 FF C9 DC 10 D3 00 DD 10 :C5
2810 39 EC 62 36 06 CC 00 61 BD CD 99 36 06 EC 62 36 :D3
2820 06 CC 00 7A BD CD B1 A4 C0 E4 C0 10 27 00 0A EC :BC
2830 62 83 00 20 ED 62 16 00 04 EC 62 ED 62 39 CC 00 :10
2840 00 DD 16 DC 06 34 06 17 FC FD 32 62 10 27 00 03 :ED
2850 16 FB 08 9E 10 4F E6 84 83 00 0D 10 26 00 03 16 :5F
2860 00 7E 9E 10 4F E6 84 83 00 2C 10 27 00 03 16 FA :DE
2870 EA DC 10 C3 00 01 DD 10 17 FF 3B CC 00 01 DD 22 :A4
2880 9E 10 4F E6 84 83 00 2D 10 26 00 0C CC FF FF DD :00
2890 22 DC 10 C3 00 01 DD 10 17 FE 79 36 06 DC 22 BD :44
28A0 CD 1F DD 12 9E 10 4F E6 84 83 00 0D 10 26 00 03 :0B
28B0 16 00 2D 9E 10 4F E6 84 83 00 2C 10 27 00 03 16 :A9
28C0 FA 99 DC 10 C3 00 01 DD 10 30 8C 02 20 02 52 00 :62
28D0 34 10 17 FE F8 32 62 10 27 00 05 CC 00 01 DD 16 :E1
28E0 17 FA 86 CC 00 00 9E 06 E7 88 1E CC EF 00 9E 06 :F3
28F0 ED 88 12 DC 14 9E 06 E7 88 17 DC 06 34 06 17 09 :DD
--------------------------------------------------------
Sum: 76 A3 FF 44 4F 24 2E CA EA 04 51 A7 E4 5A 13 3F :3D

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2900 97 32 62 9E 06 4F E6 88 11 10 27 00 03 16 FA A0 :87
2910 9E 06 4F E6 0B DD 24 DC 24 83 00 02 10 26 00 A4 :44
2920 17 08 43 9E 0E EC 84 DD 26 9E 0E EC 02 DD 28 DC :FC
2930 1A 10 27 00 0F CC A0 00 93 28 DD 12 CC A0 00 DD :BF
2940 2A 16 00 06 DC 28 D3 12 DD 2A DC 0E C3 00 04 DD :C4
2950 0E CC 00 01 DD 2C DC 2C 93 26 10 2E 00 27 DC 2A :10
2960 1F 89 4F 83 00 CC 10 2D 00 03 16 FA 2E 17 07 F6 :D8
2970 9E 2A E7 84 DC 2C C3 00 01 DD 2C DC 2A C3 00 01 :D2
2980 DD 2A 16 FF D1 17 07 DE 17 07 DB 17 07 D8 17 07 :F6
2990 D5 36 06 CC 01 00 BD CD 1F 36 06 17 07 CB E3 C1 :4D
29A0 DD 2E DC 2E 83 00 00 10 26 00 04 DC 28 DD 2E DC :BD
29B0 2E D3 12 DD 2E DC 16 83 00 01 10 26 00 03 16 07 :EA
29C0 98 16 01 06 DC 24 36 06 CC 00 00 BD CD B9 36 06 :3C
29D0 9E 06 4F E6 0C 36 06 CC 00 00 BD CD B9 A4 C0 E4 :78
29E0 C0 10 27 00 DA 17 07 7E DC 0E C3 00 02 DD 0E CC :D3
29F0 00 33 DD 18 9E 18 EC 84 DD 2A DC 2A DD 30 DC 00 :44
--------------------------------------------------------
Sum: 0E A5 AF 0A A6 AC B9 BE 40 FF 91 F6 97 A4 27 5C :B9

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2A00 CC 00 02 DD 00 DC 00 9E 0E E6 8B 4F 83 00 FE 10 :7B
2A10 26 00 16 0C 01 DC 00 9E 0E E6 8B 4F 36 06 CC 00 :99
2A20 0F A4 C0 E4 C0 D3 00 DD 00 0C 01 DC 00 9E 0E E6 :42
2A30 8B 4F 83 00 00 26 CE 0C 01 DC 00 DD 26 17 07 26 :81
2A40 9E 2A E7 84 17 07 1F 9E 2A E7 01 9E 2A EC 84 DD :35
2A50 28 DC 2A DD 32 DC 2A C3 00 02 DD 2A CC 00 03 DD :BB
2A60 2C DC 2C 93 26 10 2E 00 18 17 06 FA 9E 2A E7 84 :BD
2A70 DC 2A C3 00 01 DD 2A DC 2C C3 00 01 DD 2C 16 FF :BB
2A80 E0 DC 2A 9E 32 ED 84 17 06 DC 9E 2A E7 84 17 06 :70
2A90 D5 9E 2A E7 01 9E 2A EC 84 93 28 DD 26 DC 26 83 :00
2AA0 00 00 2C A7 DC 2A C3 00 02 DD 2A CC 00 33 DD 28 :A9
2AB0 9E 28 DC 30 ED 81 DC 2A ED 81 9F 28 16 00 0B CC :68
2AC0 00 11 9E 1C E7 88 11 16 F8 E6 16 F6 66 DC 06 34 :C7
2AD0 06 17 FA 73 32 62 10 27 00 03 16 F8 7E 9E 10 4F :E1
2AE0 E6 84 83 00 0D 10 26 00 37 CC 00 33 DD 18 9E 18 :11
2AF0 EC 84 DD 2A 9E 2A EC 84 DD 28 9E 18 EC 02 83 00 :DB
--------------------------------------------------------
Sum: 7C D1 AF D6 F1 DB EF 50 10 21 54 4E 20 24 BF 71 :24

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2B00 01 DD 32 DC 2A C3 00 02 DD 2A DC 28 DD 2E CC 00 :BD
2B10 00 DD 24 CC FF FF DD 34 CC FF FF DD 36 16 00 63 :32
2B20 DC 10 C3 00 01 DD 10 17 FB EA DD 28 9E 10 4F E6 :81
2B30 84 83 00 2C 10 27 00 03 16 FB 20 DC 10 C3 00 01 :4B
2B40 DD 10 17 FB CF DD 32 DC 28 DD 2A CC 00 02 DD 24 :B7
2B50 CC 00 00 DD 34 9E 10 4F E6 84 83 00 0D 10 26 00 :0A
2B60 07 DC 28 DD 2E 16 00 1B 9E 10 4F E6 84 83 00 2C :5D
2B70 10 27 00 03 16 F7 E4 DC 10 C3 00 01 DD 10 17 FB :DA
2B80 93 DD 2E 17 F7 E3 DC 32 93 2A C3 00 01 DD 26 DC :FD
2B90 26 83 00 00 10 2E 00 03 16 F8 00 DC 24 83 00 02 :7D
2BA0 10 26 00 04 DC 26 DD 36 CC 00 01 9E 06 E7 88 1E :4D
2BB0 DC 14 9E 06 E7 88 17 CC EF 00 9E 06 ED 88 12 DC :DC
2BC0 06 34 06 17 06 D2 32 62 9E 06 4F E6 88 11 10 27 :6C
2BD0 00 16 9E 06 4F E6 88 11 83 00 12 10 26 00 06 16 :6F
2BE0 F5 51 16 00 03 16 F7 C8 DC 34 9E 0E E7 84 DC 0E :45
2BF0 C3 00 01 DD 0E DC 36 9E 0E ED 84 DC 0E C3 00 02 :BD
--------------------------------------------------------
Sum: 84 95 DF A7 B1 B7 CA 82 E5 88 B9 1C EA E3 E7 BA :03

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
```

```
2C00 DD 0E DC 28 9E 0E ED 84 DC 0E C3 00 02 DD 0E CC :72
2C10 00 01 DD 2C DC 2C 93 26 10 2E 00 1D 9E 2A 4F E6 :23
2C20 84 34 06 17 05 7C 32 62 DC 2A C3 00 01 DD 2A DC :97
2C30 2C C3 00 01 DD 2C 16 FF DB DC 24 83 00 02 10 26 :A4
2C40 00 53 CC 00 FF 34 06 17 05 58 32 62 CC 00 00 34 :60
2C50 06 17 05 4E 32 62 CC 00 00 34 06 17 05 44 32 62 :FE
2C60 DC 2E 36 06 CC 01 00 BD CD 3B 34 06 17 05 33 32 :93
2C70 62 DC 2E 36 06 DC 2E 36 06 CC 01 00 BD CD 3B 36 :B6
2C80 06 CC 01 00 BD CD 1F A3 C1 53 43 C3 00 01 34 06 :74
2C90 17 05 0F 32 62 CC 00 1A 34 06 17 05 05 32 62 DC :70
2CA0 24 83 00 00 10 26 00 14 CC 00 00 34 06 17 04 F2 :04
2CB0 32 62 CC 00 00 34 06 17 04 E8 32 62 DC 06 34 06 :4D
2CC0 17 05 16 32 62 9E 06 4F E6 88 11 10 27 00 03 16 :88
2CD0 F8 10 16 F4 5E DC 06 DD 1C DC 06 34 06 17 F8 67 :DD
2CE0 32 62 10 27 00 11 CC 00 2A 9E 06 E7 84 CC 00 2A :D7
2CF0 9E 06 E7 08 16 00 13 9E 06 4F E6 08 83 00 00 10 :30
--------------------------------------------------------
Sum: 23 AD F3 7D 64 D3 D8 C7 72 67 A6 B0 61 2F 00 43 :18

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2D00 26 00 07 CC 00 2A 9E 06 E7 08 17 F6 5C DC 14 9E :AD
2D10 06 ED 88 16 DC 06 36 06 CC 01 01 34 06 37 10 34 :32
2D20 10 17 0B EA 32 64 9E 06 4F E6 88 11 10 27 00 03 :5E
2D30 16 F6 7D CC 01 04 9E 06 ED 88 14 CC EF 00 9E 06 :E6
2D40 ED 88 12 CC 00 04 DD 38 CC 00 00 DD 3A CC 00 00 :1B
2D50 DD 3C BD CD 17 DC 06 34 06 17 0E 07 32 62 9E 06 :3A
2D60 4F E6 88 11 10 27 00 03 16 F6 45 CC 00 00 DD 3E :40
2D70 DC 3E 83 00 E0 10 2E 01 DA DC 3E 9E 0E E6 8B 4F :1C
2D80 83 00 00 10 26 00 03 16 01 BF DC 3E 9E 0E E6 8B :C9
2D90 4F 83 00 FF 10 26 00 03 16 02 04 0F 03 DC 02 36 :4C
2DA0 06 CC 00 07 BD CD B1 36 06 DC 02 9E 06 E6 8B 4F :92
2DB0 36 06 CC 00 2A BD CD C1 A4 C0 E4 C0 10 27 00 21 :DD
2DC0 DC 02 9E 06 E6 8B 4F 36 06 DC 3E D3 02 9E 0E E6 :FF
2DD0 8B 4F BD CD C1 10 27 00 03 16 01 34 0C 03 16 FF :CE
2DE0 BC CC 00 08 DD 02 DC 02 36 06 CC 00 0A BD CD B1 :9A
2DF0 36 06 DC 02 9E 06 E6 8B 4F 36 06 CC 00 2A BD CD :3A
--------------------------------------------------------
Sum: AE 5A F4 35 55 02 DA 5B 00 EB 1C D3 AA CD E9 02 :F9

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2E00 C1 A4 C0 E4 C0 10 27 00 21 DC 02 9E 06 E6 8B 4F :63
2E10 36 06 DC 3E D3 02 9E 0E E6 8B 4F BD CD C1 10 27 :19
2E20 00 03 16 00 EB 0C 03 16 FF BC 0F 03 DC 02 83 00 :57
2E30 07 10 2E 00 11 DC 3E D3 02 9E 0E E6 8B 4F BD CE :3C
2E40 DE 0C 03 16 FF E6 BD CD 0B 20 00 CC 00 0B DD 02 :50
2E50 DC 02 83 00 0A 10 2E 00 29 DC 3E D3 02 9E 0E E6 :53
2E60 8B 4F 83 00 20 10 2D 00 0F DC 3E D3 02 9E 0E E6 :4A
2E70 8B 4F BD CE DE 16 00 05 BD CD 0B 20 00 0C 03 16 :38
2E80 FF CE BD CD 0B 20 00 DC 3E C3 00 0F 9E 0E E6 8B :8B
2E90 4F 83 00 20 10 2D 00 10 DC 3E C3 00 0F 9E 0E E6 :BD
2EA0 8B 4F BD CE DE 16 00 05 BD CD 0B 20 00 CC 00 41 :20
2EB0 DD 18 DC 3E C3 00 0B 9E 0E E6 8B 4F 83 00 02 10 :DE
2EC0 26 00 0B CC 00 4D DD 18 16 00 33 DC 3E C3 00 0B :6D
2ED0 9E 0E E6 8B 4F 36 06 CC 00 00 BD CD B9 36 06 DC :CF
2EE0 3E C3 00 0C 9E 0E E6 8B 4F 36 06 CC 00 00 BD CD :0B
2EF0 B9 A4 C0 E4 C0 10 27 00 05 CC 00 42 DD 18 BD CD :8A
--------------------------------------------------------
Sum: 3F 96 AA 46 FF 1A 19 C7 57 1C 44 0B 42 D1 4D 6B :4B

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2F00 0B 20 20 00 DC 18 BD CE DE BD CD 0B 20 3A 20 00 :B7
2F10 DC 3E C3 00 0E 9E 0E E6 8B 4F DD 40 0F 03 DC 40 :A2
2F20 C3 00 05 9E 42 E6 8B 4F 83 00 C0 10 2C 00 11 0C :04
2F30 03 DC 40 C3 00 05 9E 42 E6 8B 4F DD 40 16 FF DE :97
2F40 DC 3C D3 02 C3 00 01 DD 3C DC 3E C3 00 20 DD 3E :E2
2F50 16 FE 1D DC 38 C3 00 01 DD 38 DC 38 83 00 11 10 :D6
2F60 2F 00 0A CC 00 01 DD 38 CC 00 01 DD 3A DC 3A 36 :4B
2F70 06 CC 00 01 BD CD B9 36 06 DC 38 36 06 CC 00 04 :72
2F80 BD CD A9 A4 C0 E4 C0 10 27 00 03 16 00 11 DC 38 :B0
2F90 9E 06 E7 88 15 DC 3A 9E 06 E7 88 16 16 FD B6 BD :ED
2FA0 CD 17 CC 00 98 93 3C BD CE B0 BD CD 0B 20 43 6C :B6
2FB0 75 73 74 65 72 73 20 46 72 65 65 00 BD CD 17 16 :FF
2FC0 F1 6A DC 06 34 06 17 F5 7E 32 62 10 27 00 03 16 :E5
2FD0 F3 89 17 F3 94 DC 14 9E 06 E7 88 17 DC 06 DD 1C :0F
2FE0 17 0A 37 10 27 00 03 16 F3 C6 30 8C 02 20 03 41 :83
2FF0 53 00 34 10 17 F7 D6 32 62 83 00 00 10 26 00 03 :CB
--------------------------------------------------------
Sum: BF 9A 50 B6 C9 D1 E5 1D 03 E5 D3 F2 51 62 03 9F :FD

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3000 16 F3 58 DC 06 34 06 17 F5 3D 32 62 10 27 00 03 :94
3010 16 F3 48 9E 06 4F E6 88 17 93 14 10 27 00 03 16 :C0
3020 F3 39 0F 01 DC 00 83 00 07 10 2E 00 19 DC 00 58 :2D
3030 49 9E 08 30 8B 36 10 DC 00 9E 06 58 49 EC 8B ED :75
3040 D1 0C 01 16 FF DE 17 09 D1 83 00 00 10 26 00 0B :86
3050 CC 00 A0 02 06 E7 88 11 16 F3 55 CC 04 3D DD 10 :4C
3060 DC 06 34 06 17 F4 E0 32 62 17 09 AE 0F 03 DC 02 :59
3070 83 00 0A 10 2E 00 18 DC 3E D3 02 9E 0E 30 8B 36 :6F
3080 10 DC 02 9E 08 E6 8B 4F E7 D1 0C 03 16 FF DF DC :EB
3090 3E C3 00 0F 9E 0E 30 8B 36 10 9E 08 4F E6 0F E7 :8E
30A0 D1 DC 06 34 06 17 0A AA 32 62 9E 06 4F E6 88 11 :BE
30B0 10 27 00 03 16 F2 F9 16 F0 79 DC 06 34 06 17 F4 :E1
30C0 86 32 62 10 27 00 03 16 F2 91 17 F2 9C DC 14 9E :20
30D0 06 ED 88 16 DC 06 36 06 CC 01 01 34 06 37 10 34 :32
30E0 10 17 08 2A 32 64 DC 06 34 06 17 04 86 32 62 9E :DE
30F0 06 4F E6 88 11 36 06 CC 00 00 BD CD C1 36 06 9E :01
--------------------------------------------------------
Sum: 35 F6 76 95 C5 0F EF 2B CB 32 EA F0 9B D1 EB 87 :D9
```

リスト1（その3）

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3100 06 4F E6 88 11 36 06 CC 00 12 BD CD C1 A4 C0 E4 :81
3110 C0 10 27 00 03 16 F2 98 16 F0 18 9E 10 4F E6 84 :1F
3120 83 00 20 10 27 00 0B CC 04 3D DD 10 17 F6 87 16 :89
3130 F1 6D 17 F5 DF DD 2E DC 2E 83 00 00 10 26 00 03 :1A
3140 16 F2 58 17 F6 70 9E 10 4F E6 84 83 00 2C 10 26 :29
3150 00 07 DC 10 C3 00 01 DD 10 34 7F DC 2E 1F 01 AD :2E
3160 84 35 7F 16 EF C6 DC 0E 83 EF 00 83 00 FF 10 2F :20
3170 00 1D DC 06 34 06 17 05 2F 32 62 9E 06 4F E6 88 :79
3180 11 10 27 00 05 32 62 16 F3 75 CC EF 00 DD 0E 9E :A3
3190 0E 4F E6 84 DD 18 DC 0E C3 00 01 DD 0E DC 18 DD :26
31A0 18 39 EC 62 9E 0E E7 84 DC 0E C3 00 01 DD 0E DC :2B
31B0 0E 83 EF 00 83 00 FF 10 2F 00 1D DC 06 34 06 17 :91
31C0 05 CE 32 62 9E 06 4F E6 88 11 10 27 00 05 32 64 :AB
31D0 16 F3 49 CC EF 00 DD 0E 39 DC 0E 83 EF 00 10 27 :C4
31E0 00 5F 9E 06 4F E6 88 14 83 00 27 10 2F 00 09 CC :92
31F0 00 13 9E 06 E7 88 11 39 DC 0E 83 EF 00 DD 02 CC :77
---------------------------------------------------------
Sum: 34 65 72 F0 BC 31 AC 05 3A 7B 8C 4C 5F 54 BB 9C :30
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3200 EF 00 DD 0E DC 02 83 00 00 10 27 00 12 DC 02 9E :00
3210 0E 30 8B 36 10 CC 00 00 E7 D1 0C 03 16 FF E5 DC :78
3220 06 34 06 17 09 2C 32 62 9E 06 4F E6 88 11 10 27 :C9
3230 00 01 39 9E 06 4F E6 88 18 C3 00 01 9E 06 E7 88 :8A
3240 18 DC 24 9E 06 E7 0B CC 00 00 9E 06 E7 0C DC 06 :F3
3250 34 06 17 01 DF 32 62 39 34 7F 17 00 33 34 10 17 :56
3260 00 36 32 62 35 FF 34 7F 17 00 25 34 10 17 01 C4 :0D
3270 32 62 35 FF 34 7F 17 00 17 34 10 17 04 2A 32 62 :C6
3280 35 FF 34 7F 17 00 09 34 10 17 05 04 32 62 35 FF :33
3290 C6 CC 1F 9B CE CC 00 39 EC 62 DD 1C CC 00 00 9E :D0
32A0 1C E7 88 1F CC 00 00 9E 1C E7 88 18 CC 00 00 9E :21
32B0 1C E7 88 16 9E 1C EC 88 12 DD 44 DC 1C 36 06 CC :02
32C0 01 01 34 06 37 10 34 10 17 06 43 32 64 9E 1C 4F :C6
32D0 E6 88 11 10 27 00 03 16 00 BC 9E 1C 4F E6 88 1E :20
32E0 10 27 00 03 16 00 B7 17 07 30 10 27 00 03 16 00 :A5
32F0 A5 DC 3E C3 00 0F 9E 0E E6 8B 4F 83 00 57 10 26 :0D
---------------------------------------------------------
Sum: 50 04 2F 24 0C E7 D4 4C 2D 17 5A 47 15 E9 02 06 :A5
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3300 00 09 CC 00 17 9E 1C E7 88 11 39 0F 03 DC 02 83 :D2
3310 00 04 10 2E 00 1E CC 00 0B D3 02 9E 1C 30 8B 36 :B7
3320 10 DC 3E D3 02 C3 00 0B 9E 0E E6 8B 4F E7 D1 0C :FD
3330 03 16 FF D9 0F 03 DC 02 83 00 0F 10 2E 00 1E CC :9B
3340 00 20 D3 02 9E 1C 30 8B 36 10 DC 3E D3 02 C3 00 :62
3350 10 9E 0E E6 8B 4F E7 D1 0C 03 16 FF D9 9E 1C 4F :3A
3360 E6 0E DD 46 DC 46 C3 00 05 9E 1C E7 88 19 CC 00 :0F
3370 01 9E 1C E7 88 1F CC EF 00 C3 01 00 DD 0E 17 05 :CF
3380 47 DC 48 9E 1C E7 88 14 DC 38 9E 1C E7 88 15 DC :D6
3390 3A 9E 1C E7 88 16 DC 44 9E 1C ED 88 12 39 DC 1C :0B
33A0 34 06 17 01 CE 32 62 10 27 00 03 16 FF E8 9E 1C :A5
33B0 4F E6 88 11 83 00 10 10 26 00 0A 17 05 C2 10 27 :B6
33C0 00 03 16 FF D1 CC 00 05 DD 02 DC 02 83 00 9C 10 :A6
33D0 2E 00 17 DC 02 9E 42 E6 8B 4F 1D 83 FF FF 10 26 :97
33E0 00 03 16 00 10 0C 03 16 FF E0 CC 00 13 9E 1C E7 :AD
33F0 88 11 16 FF A1 DC 02 9E 1C E7 88 19 DC 02 83 00 :D0
---------------------------------------------------------
Sum: C4 E6 4F 60 2E D3 87 56 45 D2 24 DB 1B C4 28 3D :91
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3400 05 DD 46 DC 46 9E 1C E7 0E CC 00 00 9E 1C E7 88 :EE
3410 11 CC 00 02 9E 1C E7 88 1F CC EF 00 DD 0E DC 42 :EB
3420 C3 00 01 DD 18 9E 18 EC 84 DD 4A 9E 18 EC 02 DD :87
3430 4C 16 FF 4A EC 62 DD 1C 9E 1C EC 88 12 DD 44 9E :F1
3440 1C 4F E6 88 1F 83 00 00 10 26 00 09 CC 00 16 9E :3A
3450 1C E7 88 11 39 9E 1C 4F E6 88 1F 83 00 02 10 26 :26
3460 01 06 9E 1C 4F E6 88 19 DD 4E 9E 1C 4F E6 88 18 :51
3470 DD 50 DC 50 83 00 00 10 27 00 07 DC 50 83 00 01 :CA
3480 DD 50 DC 4E 9E 42 30 8B 36 10 DC 50 C3 00 C0 E7 :CE
3490 D1 CC EF 00 DD 0E CC EF 00 9E 1C ED 88 12 9E 1C :2D
34A0 EC 88 1A 9E 1C ED 88 14 9E 1C 4F E6 88 1C 9E 1C :1E
34B0 E7 88 16 9E 1C 4F E6 88 1D DD 3E DC 1C 34 06 17 :7D
34C0 06 A1 32 62 9E 1C 4F E6 88 11 10 27 00 03 16 FE :11
34D0 C5 0F 03 DC 02 83 00 0F 10 2E 00 18 DC 3E D3 02 :8C
34E0 9E 0E 30 8B 36 10 DC 02 9E 1C E6 8B 4F E7 D1 0C :C9
34F0 03 16 FF DF 0F 01 CC 00 10 DD 02 DC 02 83 00 1F :42
---------------------------------------------------------
Sum: 28 4B 8D 3C AA FD FD FC 80 6C 66 4F 2C 6B 73 83 :0A
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3500 10 2E 00 1A DC 3E D3 02 9E 0E 30 8B 36 10 DC 00 :D0
3510 9E 0A E6 8B 4F E7 D1 0C 01 0C 03 16 FF DD DC 1C :26
3520 34 06 17 06 2D 32 62 9E 1C 4F E6 88 11 10 27 00 :D7
3530 03 16 FE 62 DC 42 C3 00 01 DD 18 DC 4A 36 06 9E :50
3540 18 EC 84 BD CD C1 36 06 DC 4C 36 06 9E 18 EC 02 :17
3550 BD CD C1 AA C0 EA C0 10 27 00 03 16 EF DC CC 01 :47
3560 01 34 06 17 03 FE 32 62 CC 00 00 9E 1C E7 88 1F :FB
3570 16 FE 23 EC 62 DD 1C 17 04 A0 9E 1C 4F E6 88 11 :C1
3580 10 27 00 16 9E 1C 4F E6 88 11 83 00 10 10 26 00 :9E
3590 06 16 01 08 16 00 03 16 01 08 DC 3E C3 00 0F 9E :E7
35A0 0E E6 8B 4F 83 00 52 10 26 00 09 CC 00 17 9E 1C :7F
35B0 E7 88 11 39 BD CD 17 BD CD 0B 41 72 65 20 79 6F :0F
35C0 75 20 73 75 72 65 28 59 20 6F 72 20 4E 29 3F 00 :AC
35D0 17 F1 C4 9E 10 4F E6 84 34 06 17 F2 34 32 62 DD :1B
35E0 18 DC 18 83 00 4E 10 26 00 09 CC 00 12 9E 1C E7 :9B
35F0 88 11 39 DC 18 83 00 59 10 27 00 03 16 FF 9B DC :68
---------------------------------------------------------
Sum: 08 E8 8E 8F B4 8D E6 60 6F FB 06 6C 6A 33 51 B6 :14
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3600 3E 9E 0E 30 8B 36 10 CC 00 00 E7 D1 DC 1C 34 06 :A1
3610 17 05 3F 32 62 9E 1C 4F E6 88 11 10 27 00 03 16 :C7
3620 00 80 DC 3E C3 00 0E 9E 0E E6 8B 4F DD 02 DC 3E :D0
3630 C3 00 0E 9E 0E 30 8B 36 10 CC 00 FF E7 D1 DC 02 :DF
3640 C3 00 05 9E 42 E6 8B 4F 83 00 C0 10 2C 00 23 DC :E6
3650 02 C3 00 05 9E 42 E6 8B 4F DD 18 DC 02 C3 00 05 :05
3660 9E 42 30 8B 36 10 CC 00 FF E7 D1 DC 18 DD 02 16 :4D
3670 FF CC DC 02 C3 00 05 9E 42 30 8B 36 10 CC 00 FF :1D
3680 E7 D1 CC 01 01 34 06 17 02 DA 32 62 9E 1C 4F E6 :36
3690 88 11 83 00 00 10 27 00 03 16 00 06 CC 00 00 DD :1B
36A0 18 39 CC 00 01 DD 18 39 EC 62 DD 1C 9E 1C 4F E6 :82
36B0 88 1F 83 00 01 10 27 00 09 CC 00 16 9E 1C E7 88 :76
36C0 11 39 9E 1C 4F E6 88 14 DD 48 9E 1C 4F E6 88 15 :86
36D0 DD 38 9E 1C 4F E6 88 16 DD 3A 9E 1C 4F E6 88 19 :49
36E0 DD 4E DC 4E 9E 42 E6 8B 4F DD 46 9E 1C 4F E6 88 :8F
36F0 18 DD 50 DC 48 83 00 27 10 2F 00 09 CC 00 13 9E :D8
---------------------------------------------------------
Sum: 6C CA 4E D1 1E FE 69 93 2A DA 48 A6 49 CA A2 D7 :EB
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3700 1C E7 88 11 39 CC 00 08 DD 52 DC 46 83 00 C0 10 :4D
3710 2D 00 0A DC 46 83 00 C0 C3 00 01 DD 52 DC 50 93 :4E
3720 52 10 2D 00 09 CC 00 15 9E 1C E7 88 11 39 DC 1C :E4
3730 34 06 17 04 2E 32 62 9E 1C 4F E6 88 11 10 27 00 :D6
3740 01 39 17 01 50 DC 50 C3 00 01 DD 50 DC 46 83 00 :64
3750 C0 10 2C 00 1E DC 50 93 52 10 2D 00 16 CC 00 00 :4A
3760 DD 50 DC 46 C3 00 05 DD 4E DC 4E 9E 1C E7 88 19 :AE
3770 17 01 55 DC 48 9E 1C E7 88 14 DC 38 9E 1C E7 88 :0B
3780 15 DC 3A 9E 1C E7 88 16 DC 50 9E 1C E7 88 18 39 :10
3790 EC 62 DD 1C 9E 1C 4F E6 88 1F 83 00 02 10 27 00 :99
37A0 22 9E 1C 4F E6 88 1F 83 00 01 10 26 00 0C CC 00 :4A
37B0 00 9E 1C E7 88 11 39 16 00 09 CC 00 16 9E 1C E7 :15
37C0 88 11 39 9E 1C 4F E6 88 14 DD 48 9E 1C 4F E6 88 :F9
37D0 15 DD 38 9E 1C 4F E6 88 16 DD 3A 9E 1C 4F E6 88 :45
37E0 19 DD 4E 9E 1C 4F E6 88 18 DD 50 DC 48 83 00 27 :CE
37F0 10 2F 00 09 CC 00 13 9E 1C E7 88 11 39 DC 1C 34 :C6
---------------------------------------------------------
Sum: 6D 0B 58 E7 77 2C 17 60 44 B5 35 C4 5B 79 14 EB :96
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3800 06 17 03 4E 32 62 9E 1C 4F E6 88 11 10 27 00 01 :C2
3810 39 17 00 81 DC 50 C3 00 01 DD 50 DC 50 83 00 08 :A5
3820 10 2D 00 54 CC 00 00 DD 50 CC 00 05 DD 02 DC 02 :18
3830 83 00 9C 10 2E 00 1F DC 02 93 4E 10 27 00 12 DC :60
3840 02 9E 42 E6 8B 4F 1D 83 FF FF 10 26 00 03 16 00 :8F
3850 0A 0C 03 16 FF D8 CC 00 9D DD 02 DC 02 9E 1C E7 :CD
3860 88 19 DC 02 83 00 05 DD 46 DC 4E 9E 42 30 8B 36 :25
3870 10 DC 46 E7 D1 17 00 50 DC 48 9E 1C E7 88 14 DC :8E
3880 38 9E 1C E7 88 15 DC 3A 9E 1C E7 88 16 DC 50 9E :95
3890 1C E7 88 18 39 DC 38 C3 00 01 DD 38 DC 38 83 00 :60
38A0 11 10 2D 00 22 CC 00 01 DD 38 DC 3A 83 00 01 10 :FC
38B0 26 00 0F DC 48 C3 00 01 DD 48 CC 00 00 DD 3A 16 :3B
38C0 00 05 CC 00 01 DD 3A 39 DC 46 47 56 47 56 DD 54 :AF
38D0 DC 54 C3 00 02 DD 48 DC 46 36 06 DC 54 58 49 58 :A1
38E0 49 A3 C1 53 43 C3 00 01 58 49 58 49 58 49 C3 00 :AD
38F0 01 DD 38 CC 00 00 DD 3A DC 38 83 00 11 10 2D 00 :DE
---------------------------------------------------------
Sum: 27 68 6E 12 57 ED E1 D4 0E BC B8 33 08 FD E3 50 :F5
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3900 0C DC 38 83 00 10 DD 38 CC 00 01 DD 3A 39 EC 62 :33
3910 DD 1C CC EE 30 DD 56 CC EC 00 DD 42 9E 1C 4F E6 :DC
3920 88 17 10 27 00 08 CC EC 00 C3 01 00 DD 42 EC 64 :C9
3930 9E 56 ED 88 14 DC 42 9E 56 ED 88 12 9E 1C EC 88 :44
3940 16 9E 56 ED 88 16 DC 56 34 06 17 02 83 32 62 DC :0D
3950 56 34 06 17 02 0D 32 62 9E 56 4F E6 88 11 9E 1C :C6
3960 E7 88 11 39 EC 62 9E 56 ED 88 14 DC 56 34 06 17 :07
3970 01 E0 32 62 9E 56 4F E6 88 11 9E 1C E7 88 11 39 :AA
3980 17 01 A7 DC 1C 34 06 17 01 D9 32 62 9E 1C 4F E6 :65
3990 88 11 10 27 00 06 CC 00 01 DD 18 39 CC 00 00 DD :7A
39A0 3E DC 3E 83 00 E0 10 2E 00 35 DC 3E 9E 0E E6 8B :65
39B0 4F 36 06 CC 00 00 BD CD B9 36 06 DC 3E 9E 0E E6 :82
39C0 8B 4F 36 06 CC 00 FF BD CD B9 AA C0 EA C0 10 27 :6F
39D0 00 03 16 01 30 DC 3E C3 00 20 DD 3E 16 FF C2 17 :50
39E0 FE B3 DC 3A 36 06 CC 00 01 BD CD B9 36 06 DC 38 :63
39F0 36 06 CC 00 04 BD CD A9 A4 C0 E4 C0 10 27 00 09 :87
---------------------------------------------------------
Sum: 4E CE 8F 52 AA 65 B1 BD 82 1C E3 3D 27 66 1B 2F :0F
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3A00 CC 00 14 9E 1C E7 88 11 39 DC 38 9E 1C E7 88 15 :A5
3A10 DC 3A 9E 1C E7 88 16 16 FF 69 17 01 0D DC 1C 34 :24
3A20 06 17 01 3F 32 62 9E 1C 4F E6 88 11 10 27 00 06 :B6
3A30 CC 00 01 DD 18 39 CC 00 00 DD 3E DC 3E 83 00 E0 :5F
3A40 10 2E 00 86 DC 3E 9E 0E E6 8B 4F 83 00 00 10 26 :03
3A50 00 03 16 00 6B DC 3E 9E 0E E6 8B 4F 83 00 FF 10 :9C
3A60 26 00 0B CC 00 10 9E 1C E7 88 11 16 00 97 0F 03 :06
3A70 DC 02 83 00 0B 10 2C 00 21 DC 02 9E 1C E6 8B 4F :21
3A80 36 06 DC 3E D3 02 9E 0E E6 8B 4F BD CD C1 10 27 :19
3A90 00 03 16 00 2B 0C 03 16 FF D6 9E 1C 4F E6 0F 36 :72
3AA0 06 DC 3E C3 00 0F 9E 0E E6 8B 4F BD CD C1 10 27 :E0
3AB0 00 03 16 00 0B CC 00 00 9E 1C E7 88 11 16 00 45 :85
3AC0 DC 3E C3 00 20 DD 3E 16 FF 71 17 FD CB DC 3A 36 :C6
3AD0 06 CC 00 01 BD CD B9 36 06 DC 38 36 06 CC 00 04 :72
3AE0 BD CD A9 A4 C0 E4 C0 10 27 00 09 CC 00 14 9E 1C :15
3AF0 E7 88 11 39 DC 38 9E 1C E7 88 15 DC 3A 9E 1C E7 :C2
---------------------------------------------------------
Sum: 4E CB 1B 07 21 F3 42 B5 FF BA 92 0B 18 C2 70 BD :A3
```

リスト1（その3）

```
3AC0 DC 3E C3 00 20 DD 3E 16 FF 71 17 FD C8 DC 3A 36 :C6
3AD0 06 CC 00 01 BD CD B9 36 06 DC 38 36 06 CC 00 04 :72
3AE0 BD CD A9 A4 C0 E4 C0 10 27 00 09 CC 00 14 9E 1C :15
3AF0 E7 88 11 39 DC 38 9E 1C E7 88 15 DC 3A 9E 1C E7 :C2
------------------------------------------------------
Sum: 4E CB 1B 07 21 F3 42 B5 FF BA 92 0B 18 C2 70 BD :A3


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3B00 88 16 16 FF 18 9E 1C EC 88 14 9E 1C ED 88 1A 9E :F4
3B10 1C 4F E6 88 16 9E 1C E7 88 1C DC 3E 9E 1C E7 88 :77
3B20 1D 9E 1C 4F E6 88 11 DD 18 39 CC 01 04 9E 1C ED :4B
3B30 88 14 CC 00 00 9E 1C E7 88 16 CC EF 00 9E 1C ED :09
3B40 88 12 CC 00 04 DD 38 CC 00 00 DD 3A CC EF 00 DD :FA
```

```
3B50 0E 39 EC 62 C3 00 10 DD 58 CC 00 09 9E 58 E7 84 :D3
3B60 16 00 0E EC 62 C3 00 10 DD 58 CC 00 0A 9E 58 E7 :2D
3B70 84 DC 58 34 06 17 00 46 32 62 9E 58 4F E6 01 10 :1F
3B80 27 00 3A 0F 03 9E 58 4F E6 01 36 06 CC 00 0C BD :70
3B90 CD 99 36 06 DC 02 36 06 CC 00 05 BD CD A1 A4 C0 :1C
3BA0 E4 C0 10 27 00 17 DC 58 34 06 17 00 34 32 62 DC :1B
3BB0 58 34 06 17 00 08 32 62 0C 03 16 FF C8 39 AE 62 :7A
3BC0 34 79 1A 40 86 FD 1F 8B 17 E4 4A 35 79 A7 01 39 :08
3BD0 EC 62 C3 00 10 DD 58 DC 58 34 06 17 00 03 32 62 :72
3BE0 39 AE 62 34 79 1A 40 86 FD 1F 8B 17 E4 24 35 79 ;4A
3BF0 39 10 DE FE 35 FF 00 00 00 00 00 00 00 00 00 00 :59
------------------------------------------------------
Sum: 3B 64 A5 1D 66 CB 00 92 75 46 9C 0A 44 85 A1 27 :16
```

リスト2　FBASICコマンドアセンブル・リスト

```
00010                                   NAM   FBASIC
00020  6000                             ORG   $6000
00030                                   OPT   MEM
00040            EE60        PARAM      EQU   $EE60
00050            0033        TTOP       EQU   $33
00060            0035        TEND       EQU   $35
00070            003B        VTOP       EQU   $3B
00080            003D        ATOP       EQU   $3D
00090            D025        DENTRY     EQU   $D025
00100  6000 20   02    6004  START      BRA   ENTRY
00110  6002 20   31    6035  START1     BRA   ENTRY1
00120            6004        ENTRY      EQU   *
00130  6004 30   8C FB                  LEAX  START1,PCR
00140  6007 AF   8C48                   STX   ENTAD+1,PCR
00150  600A 1A   50                     ORCC  #$50
00160  600C 10CE 0100                   LDS   #$100
00170  6010 4F                          CLRA
00180  6011 1F   8B                     TFR   A,DP
00190  6013 9E   33                     LDX   TTOP
00200  6015 AF   8D 0101                STX   STEXT,PCR
00210  6019 EC   84                     LDD   ,X
00220  601B ED   8D 00FF                STD   BTOP,PCR
00230  601F 9E   35                     LDX   TEND
00240  6021 AF   8D 00F7                STX   ETEXT,PCR
00250  6025 C6   11                     LDB   #17
00260  6027 30   8D 0018                LEAX  PROG,PCR
00270  602B CE   08F0                   LDU   #$8F0
00280  602E 8D   76    60A6             BSR   MOVE
00290  6030 4F                          CLRA
00300  6031 43                          COMA
00310  6032 7E   08F0                   JMP   $8F0
00320  6035 30   8D 001B     ENTRY1     LEAX  INIT,PCR
00330  6039 CE   0800                   LDU   #$800
00340  603C C6   F0                     LDB   #$F0
00350  603E 8D   66    60A6             BSR   MOVE
00360  6040 7E   0800                   JMP   $800
00370  6043 30   8D 0007     PROG       LEAX  PROG1,PCR
00380  6047 7D   FD0F                   TST   $FD0F
00390  604A 6E   9F FBFE                JMP   [$FBFE]
00400  604E 7F   FD0F        PROG1      CLR   $FD0F
00410  6051 7E   A002        ENTAD      JMP   $A002
00420  6054 7D   FD0F        INIT       TST   $FD0F
00430  6057 7F   08FF                   CLR   $8FF
00440  605A 30   8D 008E                LEAX  AREA,PCR
00450  605E CE   FC00                   LDU   #$FC00
00460  6061 C6   2E                     LDB   #STEXT-AREA
00470  6063 8D   41    60A6             BSR   MOVE
00480  6065 86   19                     LDA   #25
00490  6067 B7   030D                   STA   $030D    ;LINE NUMBER
00500  606A 4A                          DECA
00510  606B B7   030F                   STA   $030F    ;SCROLE LINE NUMBER
00520  606E 86   50                     LDA   #80
00530  6070 97   C3                     STA   $C3      ;WIDTH
00540  6072 BD   C8BC                   JSR   $C8BC    ;SET SCREEN
00550                        *  EXTEND KEYWORD
00560  6075 7C   0203                   INC   $203     ;STATEMENT NUMBER
00570  6078 108E FC00                   LDY   #$FC00
00580  607C 10BF 0204                   STY   $204     ;TABLE ADDRESS
00590  6080 31   A9 0021                LEAY  ANL-AREA,Y
00600  6084 10BF 0210                   STY   $210     ;EXECUTE ADDRESS
00610  6088 8D   24    60AE             BSR   SET
00620  608A 8E   7FFF                   LDX   #$7FFF
00630  608D BF   059D                   STX   $059D    ;RAM END
00640  6090 9F   45                     STX   $45      ;CHARACTER END
00650  6092 30   89 FED4                LEAX  -300,X
00660  6096 9F   3F                     STX   $3F      ;STACK END
00670  6098 1F   14                     TFR   X,S
00680  609A BD   8F39                   JSR   $8F39    ;NEW
00690  609D 8D   16    60B5             BSR   SET1
00700  609F 8D   27    60C8             BSR   SET2
00710  60A1 1C   AF                     ANDCC #$AF
00720  60A3 7E   863E                   JMP   $863E
00730  60A6 A6   80          MOVE       LDA   ,X+
00740  60A8 A7   C0                     STA   ,U+
```

リスト2　FBASICコマンド　アセンブル・リスト

```
00750  60AA 5A                    DECB
00760  60AB 26   F9   60A6        BNE   MOVE
00770  60AD 39                    RTS
00780  60AE EE   8D 0068   SET    LDU   STEXT,PCR
00790  60B2 DF   33               STU   TTOP
00800  60B4 39                    RTS
00810  60B5 9E   33        SET1   LDX   TTOP
00820  60B7 EC   8D 0063          LDD   BTOP,PCR
00830  60BB ED   84               STD   ,X
00840  60BD AE   8D 005B          LDX   ETEXT,PCR
00850  60C1 9F   35               STX   TEND
00860  60C3 9F   3B               STX   VTOP
00870  60C5 9F   3D               STX   ATOP
00880  60C7 39                    RTS
00890  60C8 8E   0302      SET2   LDX   #$0302
00900  60CB 108E 1B67             LDY   #$1B67
00910  60CF 8D   0B   60DC        BSR   HLT
00920  60D1 BF   FC82             STX   $FC82
00930  60D4 10BF FC84             STY   $FC84
00940  60D8 7F   FD05             CLR   $FD05
00950  60DB 39                    RTS
00960  60DC B6   FD05      HLT    LDA   $FD05
00970  60DF 2B   FB   60DC        BMI   HLT
00980  60E1 86   80               LDA   #$80
00990  60E3 B7   FD05             STA   $FD05
01000  60E6 B6   FD05      HLT1   LDA   $FD05
01010  60E9 2A   FB   60E6        BPL   HLT1
01020  60EB 39                    RTS
01030  60EC      43        AREA   FCC   'CHAIホ'
       60ED      48
       60EE      41
       60EF      49
       60F0      CE
01040  60F1      45               FCC   'ERASナ'
       60F2      52
       60F3      41
       60F4      53
       60F5      C5
01050  60F6      4C               FCC   'LLISヤ'
       60F7      4C
       60F8      49
       60F9      53
       60FA      D4
01060  60FB      4C               FCC   'LPRINヤ'
       60FC      50
       60FD      52
       60FE      49
       60FF      4E
       6100      D4
01070  6101      53               FCC   'SOUNド'
       6102      4F
       6103      55
       6104      4E
       6105      C4
01080  6106      50               FCC   'PLAル'
       6107      4C
       6108      41
       6109      D9
01090  610A      42               FCC   'BYナ'
       610B      59
       610C      C5
01100  610D 81   F4        ANL    CMPA  #$F4
01110  610F 27   03   6114        BEQ   BYE
01120  6111 7E   92A0             JMP   $92A0
01130  6114 7F   FD0F      BYE    CLR   $FD0F
01140  6117 7E   D025             JMP   DENTRY
01150  611A      0000      STEXT  FDB   0
01160  611C      0000      ETEXT  FDB   0
01170  611E      0000      BTOP   FDB   0
01180            6000             END   START
```

リスト3　KDコンパイラ変更箇所(V3.6)

Addr	old	new	Addr	old	new	Addr	old	new	Addr	old	new	Addr	old	new
2003	4C	9F	2007	4C	9F	2011	4D	CD	2014	4D	CD	202E	4D	CD
2031	4D	CD	204E	4D	CD	2051	4D	CD	2054	4D	CD	2057	4D	CD
2067	4D	CD	206A	4D	CD	206D	4D	CD	2073	42	04	2074	00	3D
2082	43	96	208E	4B	9E	209A	4B	9E	20C0	4C	9F	20CE	4C	9F
2124	4D	CD	2127	4D	CD	213B	4D	CD	213E	4D	CD	214F	4E	CE
215D	4D	CD	2170	4D	CD	2182	4D	CD	2198	4D	CD	21A2	4D	CD
21E1	4D	CD	21E4	4D	CD	21FA	4D	CD	21FD	4D	CD	2218	4E	CE
2237	4D	CD	223A	4D	CD	223D	4D	CD	225F	4E	CE	2262	4D	CD
226B	4E	CE	226E	4D	CD	2273	4D	CD	2276	4D	CD	2279	4D	CD
229A	4D	CD	22A9	4D	CD	22D3	4D	CD	2305	4D	CD	2314	4D	CD
2326	4D	CD	239B	4D	CD	23EC	4E	CE	2405	4D	CD	2430	4D	CD
243E	4D	CD	247F	4E	CE	24A5	4E	CE	24CC	4E	CE	24F2	4E	CE
2509	4D	CD	2517	4D	CD	2528	4E	CE	2586	4D	CD	2589	4D	CD
26E9	4D	CD	26F7	4D	CD	2707	4D	CD	2715	4D	CD	28C4	4D	CD
28C7	4D	CD	299E	4D	CD	29A1	4D	CD	2A67	4D	CD	2A6A	4D	CD
2AC4	4D	CD	2AC7	4D	CD	2C1D	4D	CD	2C20	4D	CD	2FA3	4D	CD
2FA6	4D	CD	3150	4D	CD	315C	4D	CD	3187	4D	CD	3193	4D	CD
3219	4D	CD	3225	4D	CD	327D	4D	CD	3280	4D	CD	33A3	4D	CD
3401	4D	CD	3443	4D	CD	344F	4D	CD	3460	4D	CD	346C	4D	CD
347C	4D	CD	3488	4D	CD	349D	4D	CD	34A9	4D	CD	35FC	4D	CD
3617	4D	CD	3632	4D	CD	364C	4D	CD	3666	4D	CD	3680	4D	CD
36E0	4D	CD	36EF	4D	CD	3807	4D	CD	3816	4D	CD	3866	4D	CD
3872	4D	CD	387E	4D	CD	388E	4D	CD	389E	4D	CD	38D2	4D	CD
38E6	4D	CD	38F2	4D	CD	38FE	4D	CD	390E	4D	CD	391E	4D	CD
3952	4D	CD	397D	4D	CD	39A3	4D	CD	39CD	4D	CD	3C8B	4F	CF
3D18	4D	CD	3D24	4D	CD	3D34	4D	CD	3D44	4D	CD	3D6C	4D	CD
3F70	4D	CD	3F7C	4D	CD	3FA5	4D	CD	3FB1	4D	CD	402F	4D	CD
4032	4D	CD	4049	4D	CD	4058	4E	CE	405B	4D	CD	405E	4D	CD
406A	4E	CE	4081	4E	CE	4090	4E	CE	4093	4D	CD	409B	4D	CD
40A1	4D	CD	40A4	4D	CD	40BC	4D	CD	40BF	4D	CD	40E0	4D	CD
40EC	4D	CD	4126	4D	CD	4137	4D	CD	4148	4D	CD	4159	4D	CD
416A	4D	CD	41F1	4D	CD	41F7	7F	FF						

リスト4　TYPEコマンド　リース・リスト

```
10 '(* TYPE V 1.0 PROGRAMMED BY ON 84.1 *)
20 'CONST EOPEN=$D018,EREAD=$D01E,EGTL=$CD06
30 'CONST BUFAD=$043D,PARMAD=$EE60,FATAD=$EC00;EOL=$0
40 'GOTO TYPE
50 'OPEN;CODE $AE62#,$BD,EOPEN#;RETURN
60 'READ;CODE $AE62#,$BD,EREAD#;RETURN
70 'FORMER;PRINT "FORMAT ";GOTO EROUT
80 'DVCK;IF DEV=0 OR DEV=1 THEN RETURN;FI;PRINT/"DEVICE ";CODE 3262#
90 'EROUT;PRINT "ERROR";GOTO LOOP1
100 'OPENER;
110 'IF FC:17)=16 THEN PRINT"File Not Found";GOTO LOOP1;FI
120 'IF FC:17)=10 THEN PRINT"Drive Not Ready";GOTO LOOP1;FI
130 'PRINT"OPEN ER CODE=",FC:17);GOTO LOOP1
140 'RDER;PRINT"READ ER CODE=",FC:17);GOTO LOOP1
```

リスト 4 TYPEコマンド　リース・リスト

```
150 '(* GTDAF[X:I)] *)
160 'GTDAF;SKIPSP[];WHILE X:0)<>'"';IF X:0)=EOL THEN %1=1;RETURN;FI;X=X+1;WEND
170 'X=X+1
180 'FCB=%1
190 'FOR I=0 TO 7;FCB:I)=' ';NEXT;FCB(4)=0;FCB:10)=0;FCB:15)=0;I=0;J=0
200 'WHILE X:0)<>'"'
210 '  IF X:0)=':' THEN DEV=FCB:0)-$30;I=0;FCB:0)=' ' ELSE
220 '  IF X:0)=' ' THEN J=J+1;I=0 ELSE
230 '  IF X:0)=EOL THEN GOTO GTDAF2;FI
240 '  IF J=0 THEN IF I<=7 THEN FCB:I)=X:0);I=I+1;FI;ELSE
250 '  IF J=1 THEN IF I<=2 THEN FCB:I+8)=X:0);I=I+1;FI;ELSE
260 '  IF J=2 THEN IF I=0 THEN FCB:15)=X:0);I=I+1;FI;FI;FI;FI;FI;FI;X=X+1;WEND
270 'GTDAF2;%1=0;RETURN
280 'SKIPSP;WHILE X:0)=' ';X=X+1;WEND;RETURN
290 '(* TYPE *)
300 'TYPE;FC=WORK[];X=PARMAD;DEV=PEEK($D024)
310 'IF GTDAF[FC] THEN GOTO FORMER;FI
320 'DVCK[]
330 'FC:30)=0;FC(9)=BUFAD
340 'FC:23)=DEV;GPOPN[FC];IF FC:17) THEN GOTO OPENER;FI
350 'PRINT /;FOR I=0 TO 7;PRINT CHR$(FC:I));NEXT;PRINT" "
360 'I=8;WHILE I<=10 AND FC:I)<>0;PRINT CHR$(FC:I));I=I+1;WEND;PRINT" "
370 'IF FC:15)<>0 THEN PRINT CHR$(FC:15));FI;PRINT /
380 'I=32;WHILE FC:I)>=' ' AND I<48;PRINT CHR$(FC:I));I=I+1;WEND;PRINT /
390 'PRINT"Directory track=",FC:26)," secter=",FC:27)," side=",FC:28),/
400 'PRINT"Start track=",FC:20)," secter=",FC:21)," side=",FC:22),/
401 'J=FC:14);FAT=FATAD;IF FC:23) THEN FAT=FAT+256;FI
402 'PRINT"Used clusters are"," $",HEX2$(J)
403 'WHILE FAT:J+5)<$C0;J=FAT:J+5);PRINT " $",HEX2$(J);WEND
410 'PRINT /;FTYP=FC:11)
420 'IF FTYP=2 THEN PRINT"*MACHINE*";GTCH[];LEN=BUF(0);SR=BUF(1);BUF=BUF+4
430 '  AR=SR;ER=AR+LEN-1;I=1
440 '  WHILE I<=LEN;A=GTCH[];I=I+1;WEND
450 '  GTCH[];GTCH[];GTCH[];XADR=GTCH[]*256+GTCH[]
460 '  DSPSUB[]
470 'ELSE IF FTYP=0 AND FC:12)=0 THEN PRINT"*BASIC*";GTCH[];BUF=BUF+2
480 '  I=0;REPEAT I=I+1;UNTIL BUF:I+3)=0 AND BUF:I+1)<>254;SR=BUF(0)-I-4;AR=SR;P
=0
490 '  REPEAT;A=GTCH[];IF A=0 THEN P=P+1;ELSE IF A<>$1A THEN P=0;FI;FI;AR=AR+1;U
NTIL A=$1A AND P=3
500 '  XADR=SR;ER=AR;DSPSUB[]
510 'ELSE IF FC:12)=255 THEN PRINT"*ASCII*";PRINT /;A=$0D;REPEAT;PRINT CHR$(A);A
=GTCH[];UNTIL A=$1A;PRINT /
520 'ELSE FCB:17)=17;GOTO OPENER;FI;FI;FI
530 'LOOP1;END
540 'DSPSUB;PRINT//" START=",HEX$(SR)," END=",HEX$(ER)," EXEC=",HEX$(XADR),/;RET
URN
550 'GPOPN;OPEN[%1];BUF=BUFAD+256;RETURN
560 'GTCH;
570 'IF BUF-BUFAD>255 THEN READ[FC];IF FC:17) THEN CODE $3262#;GOTO RDER;FI;BUF=
BUFAD;FI
580 'A=BUF:0);BUF=BUF+1;A=A;RETURN
590 'WORK;CODE $308C#,$031F#,$1039#
```

リスト 5　WIDTHコマンド　ソース・リスト

```
10 '(* WIDTH V 1.0 PROGRAMMED ON 83.12 *)
30 'CONST PARMAD=$EE60
40 'X=PARMAD;WD=DEC[X:0)]*10-DEC[X:1)]
50 'W=80;IF WD=40 THEN W=40;FI
60 'INIT[]
70 'END
80 'DEC;%1=%1-$30;RETURN
90 'HLT;CODE $B6,$FD05#,$2BFB#,$8680#,$B7,$FD05#,$B6,$FD05#,$2AFB#;ADDR=$FC82;RE
TURN
100 'SUB;POKE $FD05,0;RETURN
110 'INIT;HLT[];CPOKE ADDR,$0100#,W,25,0,25,0,1,0;SUB[];RETURN
```

リスト 6　DATEコマンド　ソース・リスト

```
10 '(* DATE V 1.0 PROGRAMMED ON 83.12 *)
20 'CONST PARMAD=$EE60,DATEAD=$EE50,EOF=0
30 'X=PARMAD;Y=DATEAD
40 'IF X:0)=EOF THEN I=0;WHILE I<=15 AND Y:I)<>EOF;PRINT CHR$(Y:I));I=I+1;WEND;E
ND
50 'FOR I=0 TO 15;Y:I)=0;NEXT
60 'I=0;WHILE I<=15 AND X:I)<>EOF;Y:I)=X:I);I=I+1;WEND
70 'END
```

リスト 7　HELPコマンド　ソース・リスト

```
10 '(* HELP V 1.0 PROGRAMMED ON 84.1 *)
20 'Q=$22
30 'PRINT/"--IODOS9 resident commands--"/
40 'PRINT/"EXEC adr"
50 'PRINT/"    execute an address"
60 'PRINT/"FILE<S><",CHR$(Q),"<dr:><file_name><",CHR$(Q),">>"
70 'PRINT/"    list out file names"
80 'PRINT/"    if file_name comtains '*',list out matched file name"
90 'PRINT/"KILL",CHR$(Q),"<dr:>file_name<",CHR$(Q),">"
```

リスト7　HELPコマンド　ソース・リスト

```
100 'PRINT/"   delete file"
110 'PRINT/"LOAD<M>",CHR$(Q),"<dr:>file_name<",CHR$(Q),"<,<offset adr><,R>>>"/
120 'PRINT/"   load BASIC or MACHINE file"
130 'PRINT/"   if R is specified then execute"
140 'PRINT/"NAME",CHR$(Q),"<dr:>old file_name",CHR$(Q),"AS",CHR$(Q),"new file_na
me<",CHR$(Q),">"
150 'PRINT/"   rename file name"
160 'PRINT/"SAVE<M>",CHR$(Q),"<dr:>file_name<",CHR$(Q),"<,start adr,end adr<,exe
c adr>>>"
170 'PRINT/"   save BASIC or MACHINE file"
180 'PRINT//"--IODOS9 control commands--"/
190 'PRINT/"dr"
200 'PRINT/"   change mother drive"
210 'PRINT/"<dr:>command<param>"
220 'PRINT/"   execute command file with parameter"
230 'PRINT/"   command file:fn is command name,ft is COM and fm is R"
240 'PRINT//"--IODOS9 trangent commands--"/
250 'PRINT/"DATE date"
260 'PRINT/"   set date within 16 characters"
270 'PRINT/"FBASIC"
280 'PRINT/"   goto F-BASIC mode"
290 'PRINT/"   back to IODOS9 using ",CHR$(Q),"BYE",CHR$(Q)
300 'PRINT/"HELP"
310 'PRINT/"   help you to use IODOS9"
320 'PRINT/"KD"
330 'PRINT/"   high memory K compiler"
340 'PRINT/"STARTUP"
350 'PRINT/"   set volume name and start up program name"
360 'PRINT/"TYPE",CHR$(Q),"<dr:>file_name<",CHR$(Q),">"
370 'PRINT/"   list out  file statistics"
380 'PRINT/"   file position,load address,execute address"
390 'PRINT/"WIDTH n"
400 'PRINT/"   change display line width"
410 'PRINT/"   n must be 40 or 80"
420 'PRINT//"--symbols--"/
430 'PRINT/"   file_name:fn< <ft>< fm>>>"
440 'PRINT/"   dr:{0|1}..drive no."
450 'PRINT/"   adr:hex number address"
460 'PRINT/
```

リスト8　START UP　ソース・リスト

```
10 '(* STARTUP V 1.0 PROGRAMMED ON 84.1 *)
20 'CONST ERST=$D012,ESRW=$D015,EGTL=$CD06
30 'CONST BUFAD=$043D
40 'GOTO STUP
50 'RES;CODE $AE62#,$3479#,$1A40#,$86FD#,$1F8B#,$BD,ERST#,$3579#,$A701#;RETURN
60 'SRW;CODE $AE62#,$3479#,$1A40#,$86FD#,$1F8B#,$BD,ESRW#,$3579#,$A701#;RETURN
70 'GTLINE;CODE $8E,BUFAD#,$BD,EGTL#;X=BUFAD;RETURN
80 'SKIPSP;WHILE X:0)=' ';X=X+1;WEND;RETURN
90 'DEVER;PRINT/"DEVICE ER";GOTO LOOP1
100 'RDER;PRINT/"DISK READ ER CODE=",FC:17);GOTO LOOP1
110 'WTER;PRINT/"DISK WRITE ER CODE=",FC:17);GOTO LOOP1
120 '(* STUP *)
130 'STUP;FC=WORK[];FOR I=0 TO 7;FC:I)=0;NEXT;FN=FC+8
140 'PRINT/"DEVICE?";DEV=INPUT
150 'IF DEV<>0 AND DEV<>1 THEN GOTO DEVER;FI
151 'PRINT/"VOLUME NAME(within 2 character)?";GTLINE[];SKIPSP[];VL=X(0)
152 'PRINT/"FILE NAME(within 8 character)?";GTLINE[];SKIPSP[]
153 'I=0;WHILE I<=7;FN:I)=$20;I=I+1;WEND
154 'I=0;WHILE I<=7 AND X:0)<>$0D;IF X:0)>=$20 THEN FN:I)=X:0);ELSE FN:I)=$20;FI
;X=X+1;I=I+1;WEND
160 'FC(1)=BUFAD;FC:5)=1;FC:7)=DEV
161 'RES[FC]
170 'FC:0)=$0A;SRW[FC];IF FC:1) THEN GOTO RDER;FI
180 'X=BUFAD+5;FOR I=0 TO 7;X:I)=FN:I);NEXT
210 'FC:0)=$09;SRW[FC];IF FC:1) THEN GOTO WTER;FI
220 'FC:4)=1
230 'FC:0)=$0A;SRW[FC];IF FC:1) THEN GOTO RDER;FI
240 'A=BUFAD+1;A(0)=VL
250 'FC:0)=$09;SRW[FC];IF FC:1) THEN GOTO WTER;FI
260 'LOOP1;END
270 'WORK;CODE $308C#,$031F#,$1039#
```

Kコンパイラ有象無象

Kコンパイラ用サブルーチン

■COMPAC S.

拡張Kコンパイラを利用する上での注意点，プログラミングの特殊なテクニックなどを説明します．拡張Kコンパイラを利用して，ひと味違ったプログラミングを楽しみましょう．

プログラミングの注意点

ソース・プログラムの大きさ

通常の場合，ソース・プログラムが，＄2000を越えると，コンパイラ本体を壊してしまいます．このときは，コンパイラ本体を裏RAMに移動する（FM-7の場合）か，コンパイラをファイル版に変更する（FM-8，11の場合）をしてください．

コンパイラを裏RAMに移動したときは，ソース・プログラムは，$41FFまで占有できます．ソース・プログラムが$41FFを越えるときは，一行入力バッファとシンボル・テーブルを裏RAMに移動すると，さらに，$4AFFまで占有できます．

BASICのCLEAR文は，ソース・プログラムがコンパイラなど他の部分を壊さないために使うので，その恐れがないときは，使わなくてもかまいません．一度設定すると，再設定されるまで有効です．

ソース・プログラムがどこまで使っているかを知るには＄35，＄36番地の内容をみればわかります（図1）．

コンパイルできるシンボルの数

コンパイルできるプログラムの変数の数の上限は，127です．また，変数，ラベル，定数（以上シンボルと総称する）の入っているテーブル（シンボル・テーブル）の大きさは，＄4300～＄4AFFまでの＄800バイトです．各変数，ラベル，定数1つあたり，12バイトを使うので，それらの総計の上限は，170です．

シンボル・テーブルが大きくなりすぎて壊れると，原因不明のエラーがでます．このときは，

①シンボルで未使用のものを削除し，重復して使える変数はなるべく何回も使う．

②テーブルの大きさを増やす．FM-7なら，裏RAMに移動することもできます．

旧バージョンでは，このテーブルの大きさが＄700バイトと少し小さかったので注意してください．

テーブルの大きさの変更は，エントリ・マップのシンボル・テーブル・アドレス（p.83）の値を変更することでできます．シンボル・テーブルをダンプし，名前とアドレス，および何バイト使っているかを出力するプログラムを，**リスト1**に，実行例を図2に示します．

図1　CLEAR 300，& H1FFFのときのメモリ・マップ

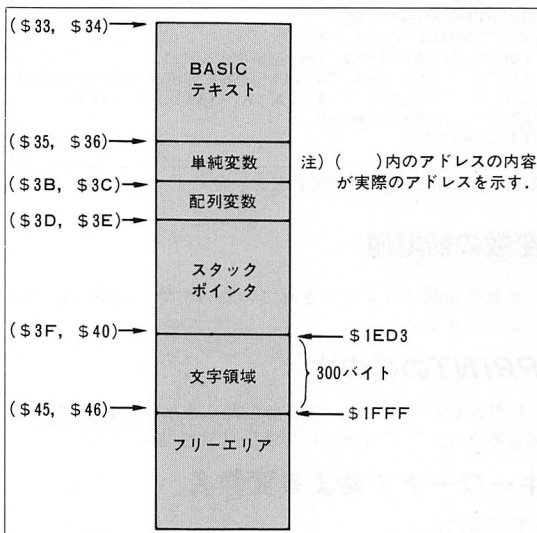



図2　シンボル・テーブル ダンプ実行例

```
K COMPILER LABEL TABLE DUMP

*** LABELS ***
CHECK      $50EC:
*** VARIABLES ***
SUM        $0000: A          $0002: I
 $0004:
*** CONSTANTS ***
LTBL       $4300:
*** DEFINES ***

60 bytes used
```

FOR-NEXT

FOR-NEXTからのGOTO文，RETURN文による飛び出しをすると，異常な動作をします．ループからの飛び出しがあるときは，FOR-NEXT文でなく，WHILE-WEND

例1　配列を0クリアするプログラム

```
10 'CONST INCI=$0C01,CLRI=$0F01
20 'I=0;FC=$6000
30 'CODE CLRI#;WHILE I<=20;FC(I)=0;CODE INCI#;WEND
```

例2　データ参照プログラム

```
10 'CONST INCI=$0C01,CLRI=$0F01
20 'I=0;FC=DATA[]
30 'CODE CLRI#;WHILE I<=9;PRINT FC:I);CODE INCI#;WEND;END
40 'DATA;CODE $308C#,$031F#,$1039#
50 'CODE 0,1,2,3,4,5,6,7,8,9,10
```

リスト1　シンボル・テーブル　ダンプ・プログラム

```
10 'CONST LTBL=$4300
20 'PRINT/"K COMPILER LABEL TABLE DUMP"/
21 'SUM=0
30 'PRINT/"*** LABELS ***"/;CHECK[0]
40 'PRINT/"*** VARIABLES ***"/;CHECK[1]
50 'PRINT/"*** CONSTANTS ***"/;CHECK[3]
60 'PRINT/"*** DEFINES ***"/;CHECK[2]
61 'PRINT/SUM*12," bytes used"/
70 'END
80 'CHECK;A=LTBL
90 '  WHILE A:0)<>$FF
100 '  IF A:0)=%1 THEN SUM=SUM+1
110 '     FOR I=4 TO 11;PRINT CHR$(A:I));NEXT
120 '      PRINT " $",HEX$(A(1)),": ";FI
130 '    A=A+12;WEND
140 'RETURN
```

文，またはREPEAT-UNTIL文を使ってください．

変数の初期値

変数の初期値はクリアされていないので，何が入っているかわかりません．

*PRINT*の後の文

PRINT文のあとに，他の文を書くときは，必ず“;”で区切ること．“␣”（空白）では区切れません．

キーワードで始まる変数名

使えません．

プログラミング技法

小さくそして速くする

0～127までの1ずつ繰り上がるカウンタは，次のようにするとプログラム・サイズも小さく，しかも実行速度があがります．

①方法

①カウンタに使う変数を決める．普通IとかJを使う．

②プログラムの先頭で，その変数に0を代入する．

③以後，0にするには，〝CODE $0F01#〃を使う．1繰り上げるには，〝CODE $0C01#〃を使う．

$6000から始まる配列FC'を，42バイト0クリアするプログラムを**例1**に示します．

*DATA*文を使う

BASICのDATA文のように，プログラム中にデータ領域をとり，読み出し，書き込みができます．

方法

①得たいアドレスの前にラベルをつけ，“CODE　$308C#，$031F#，$1039#”を書く．データ領域はこの後となる．

②このラベルに関数コールをすると，データ領域の先頭アドレスを持ち返る．

③あとは，そのアドレスをもとに，PEEK/POKE，または配列でその内容が読み書きできる．

メモリ内のデータを書き出し，プリント・アウトするプログラムを**例2**に示します．

例3　文字列長をプリント・アウトするプログラム

```
10 'N=LEN["ABCDEF"];PRINT N
20 'END
30 'LEN;A=%1;L=0
40 'WHILE A:0);A=A+1;L=L+1;WEND
50 '%1=L;RETURN
```

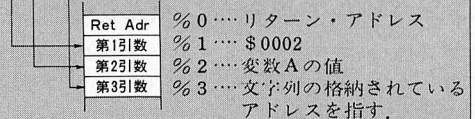
文字列を扱う

サブルーチンまたは関数に引数の文字列を書いて，それらを呼び出すと，サブルーチンまたは関数側の引数に，その文字列の格納されている先頭アドレスがセットされます．サブプログラム側では，そのアドレスを利用して文字を取り出します．なお，文字列の終わりは，0（ヌル）なので，文字長などのチェックができます．

文字列の長さをプリント・アウトするプログラムを**例3**に示します．

ローカル変数を使う

サブルーチンまたは関数のパラメータは，スタックに積まれるローカル変数です．この変数に，サブプログラム内で値を代入しても，RETURN命令とともに値は消えてしまいます．

リカーシブ・コールのとき，受け渡す値はすべてローカル変数にしなければなりません．

サブプログラムの入り口でLEAS -n,Sを，　出口でLEAS　n,Sをすると，サブプログラムの中でローカル変数を新たに増やすことができます．

サブプログラムの中で，4つのローカル変数を追加するプログラムを**例4**に示します．

プログラムの入口でLEAS -8, Sを，出口でLEAS 8, Sをしています．これによって，%変数は次のように変わります．

```
%0 ┐
%1 │
%2 ├ ローカル変数
%3 ┘
%4……リターン・アドレス（壊さない）
%5……第1パラメータ
%6……第2パラメータ
```

このプログラムは，整数を5桁のASCII文字にするものです．頭のゼロはサプレスします．

高速表示（FM-7用例5）

いろんな色のインベーダを1000匹，あっと言う間に表示します．インベーダのパターン・データを，サブの＄C100に送っておいて，それをメインからの指示で表示します．サブの表示プログラムは$C000から入っています．

SUBMITは256バイトずつ，サブに転送するルーチンです．TRANSFERはサブ表示プログラム（SUBPRO）と，表示データをサブに転送します．FPUTはサブのデータを表示させます．

```
FPUT〔X,Y,LX,LY,PAD,C〕
  X:   表示する左上のX座標（0～639）．
  Y:   表示する左上のY座標（0～199）．
  LX:  表示するデータのX方向のバイト数．
  LY:  表示するデータのY方向のドット数．
  PAD: 表示するパターン・データの入っているアドレス．＄C000からのオフセットで指定する．
       例 ＄100……＄C100からパターン・データが入っている．
  C:   カラーコード
```

参考文献

1) 吉田昌：“バトミントン・ゲーム”，I/0，'83年10月号

例4　ローカル変数を追加する

```
10 'WORK=$6000;V=123
20 'BTOD[V,WORK]
30 'FOR I=0 TO 4;PRINT CHR$(WORK:I));NEXT
40 'END
50 'BTOD;CODE $3278#;%1=1;%2=0;%3=10000;DMY=%6
60 'WHILE %2<=4;%0=%5/%3;%5=%5-%0*%3;%0=%0+$30
70 'IF %1=1 THEN IF %0=$30 THEN %0=$20;ELSE %1=0;FI;FI
80 'DMY:%2)=%0;%3=%3/10;%2=%2+1;WEND;CODE $3268#;RETURN
```

例5　高速表示（FM-7用）

```
10 'CONST PGBR=$100
20 'PAT=PATAD[];TRANSFER[PAT]
30 'FOR I=0 TO 999;X=RND(620);Y=RND(190)
31 'C=RND(6)+1
40 'FPUT[X,Y,2,8,PGBR,C]
60 'NEXT
70 'END
80 'HLT;CODE $B6,$FD05#,$2BFB#,$1A40#,$8680#,$B7,$FD05#,$B6,$FD05#,$2AFB#;ADDR=$
FC82;RETURN
90 'SUB;POKE $FD05,0;RETURN
100 '(* ************ TRANSFER ********* *)
110 'TRANSFER;B=SUBPRO[]-12;SUBMIT[B,0]
120 'SUBMIT[%1,1];RETURN
130 '(* ************ SUBMIT ********* *)
140 'SUBMIT;W=$C000+$100*%2;ADDRES=%1;FOR W1=0 TO 3;HLT[];CPOKE ADDR,$3F,'Y','A'
,'M','A','U','C','H','I',$91,$D393#,W#,$40#,$90;FOR W2=0 TO $3F;ADDR:W2)=ADDRES:
W2);NEXT;SUB[];ADDRES=ADDRES+$40;W=W+$40;NEXT;RETURN
150 '(* ********* F PUT ************ *)
160 'FPUT;W=%1+80*%2;HLT[];CPOKE ADDR,$3F,'Y','A','M','A','U','C','H','I',$91,$D
396#,$C000#,$0007#,$93,$C00C#,$90,W#,%3,%4,%5#,%6;SUB[];RETURN
170 '
180 '(* *** FPUT PROGRAM TRANSPORT *** *)
190 'SUBPRO;CODE $308C#,$031F#,$1039#
200 'CODE $34,$37,$B6,$D4
210 'CODE $09,$CC,$C0,$00,$E3,$8C,$ED,$ED
220 'CODE $8C,$ED,$1F,$01,$6F,$8C,$EC,$1A
230 'CODE $51,$69,$8C,$E7,$A6,$8C,$E4,$81
240 'CODE $08,$26,$02,$35,$B7,$A6,$8C,$D6
250 'CODE $81,$08,$24,$0C,$AE,$8C,$D0,$A5
260 'CODE $8C,$D1,$26,$04,$86,$20,$20,$02
270 'CODE $86,$21,$A7,$8C,$13,$E6,$8C,$BA
280 'CODE $4F,$E3,$8C,$B4,$1F,$02,$EC,$8C
290 'CODE $B1,$40,$ED,$8C,$B4,$E6,$8C,$B1
300 'CODE $20,$04,$A6,$80,$20,$01,$4F,$A7
310 'CODE $A5,$5C,$2D,$F4,$31,$A8,$50,$6A
320 'CODE $8C,$A0,$26,$E9,$A6,$8C,$91,$8B
330 'CODE $40,$A7,$8C,$8C,$20,$AB
340 'PATAD;CODE $308C#,$031F#,$1039#
350 'CODE $0F,$00,$7F,$E0,$FF,$F0,$E6,$70
360 'CODE $FF,$F0,$19,$80,$36,$C0,$C0,$30
```

●拡張Kコンパイラ用の強力なユーティリティ●

Kコンパイラで文字変数を使う

■石井正秀

PC-6001用の将棋対局（I/O'83年９月号）をKコンパイラ用に移植しようと思い立ちましたが，なにせ敵はBASICコンパイラ，文字変数を多用しています．そこでまず，Kでも文字変数や関数を使えるようにしようと，このサブルーチンを作りました（もっとも肝心の将棋対局はすでに'83年12月号に掲載されましたが…）．

◈概　略◈

文字変数を使うためにあたって，いろいろな方法を試みましたが，結局BASICにならってアドレス・ポインタを採用しました．文字変数には変数名がなく，すべて番号で処理します．つまり，たった１種類の配列とみなすことができます．

◈文　法◈

以下に示す引数のうち，〝Ni〟は文字変数の番号，〝NiS〟は文字変数番号または文字列，〝式〟は数式，数値を表わします．また，例はBASICの対応する形式も共に示してあります．

●LET(N1,N2S)

機能：N_1番の文字変数に，N_2番の文字変数または文字列Sを代入する．

例　LET［0，1］
　　　AS（0）=A$（1）
　LET［2，〝I/O〟］
　　　A$（2）=〝I/O〟

●ADD(N1,N2S,N3S)

機能：N_2SとN_3Sを結合し，N_1に代入する．長さが指定値（後述）を越える場合は，後ろを切り捨てる．

例）　ADD［0，1，2］
　　　A$（0）=A$（1）+A$（2）
　ADD［3，4，〝PTS.〟］
　　　A$（3）=A$（4）+〝PTS.〟

●CMP(N1S,N2S)

機能：N_1SとN_2Sを比較する．関数の値は，

　$N_1S>N_2S$　のとき　1
　$N_1S=N_2S$　のとき　0
　$N_1S<N_2S$　のとき　−1

が返される．

例）　F=CMP［0，1］
　　　F=（A$（0）=A$（1））

※ただし，BASICでは，式が成立するとき〝−1〟不立のとき〝0〟が返される．

●PRT(N1)

機能：N_1番の文字変数を画面に出力する．

例）　PRT［0］
　　　PRINT　A$（0）

●INP(N1)

機能：文字列をキーボードより入力し，それをN_1に代入する．

列）　INP［0］
　　　LINE　INPUT　A$（0）

●POI(N1)

機能：文字列のアドレスを与える．

列）　ADDR=POI［0］
　　　ADDR=VARPTR（A$（0））

※ただし，BASICではポインタのアドレスが返されるが，POIでは文字列の格納アドレスを与える．

●CHR(N1,式)

機能：式の値をキャラクタ・コードにしてN_1に代入する．

例）　CHR［0，X］　　A$（0）=CHR$（X）

●STR(N1,式)

機能：式の値の10進表現を文字列に変換して，N_1に代入する．

列）　STR［0，X］　　A$（0）=STR$（X）

●LEFT(N1,N2S,式)

機能：N_2Sの左から式の長さだけを取り出し，N_1に代入する．

例）　LEFT［0，0，3］
　　　A$(0)=LEFT$（A$（0），3）

●MIDA〔N_1, N_2S, 式$_1$, 式$_2$〕

機能：N_2Sの式$_1$の位置から，式$_2$の長さだけを取り出し，N_1に代入する．

列）　MIDA［0，0，3，2］

　　A$（0）＝MID$（AS（0），3，2）

●MIDB〔N_1, 式$_1$, 式$_2$, N_2S〕

機能：N_1の式$_1$の位置から，式$_2$の長さだけをN_2Sに置き換える．

例）：MIDB［0，3，4，1］

　　MID$（A$（0），3，4）＝A$（1）

●RIGHT〔N_1, N_2S, 式〕

機能：N_2Sの右から式の長さだけを取り出し，N_1に代入する．

例）　RIGHT［0，1，4］

　　AW（0）＝RIGHT$（A$（1），4）

●STRING〔N_1, 式$_1$, 式$_2$〕

機能：式$_2$のキャラクタを式$_1$の長さの文字列にして，N_1に代入する．

例）　STRING［0，10，$20］

　　A$（0）＝STRING$（10，&H20）

●HEX〔N_1, 式〕

機能：式の値の16進表現を，4桁の文字列に変換してN_1に代入する．

例）　HEX［0，X］

　　A$（0）＝HEX$（X）

※ただし，BASICでは桁数が一定でなく，頭の余分な〝0″は切り捨てられる．

●ASC〔N_1〕

機能：N_1の最初の1文字のコードを返す．

例）　X＝ASC［0］

　　X＝ASC（A$(0)）

●VAL〔N_1〕

機能：N_1を数値に変換する．スペースはスキップされる．文字列の頭に〝$″があれば，16進とみなす．

例）　X＝VAL［0］

　　X＝VAL(A$(0))

●LEN〔N_1〕

機能：N_1の長さを与える．

例）　X＝LEN［0］

　　X＝LEN（A$(0)）

●INSTR〔式, N_1S, N_2S〕

機能：N_1S中にN_2Sが含まれているかどうか検索する．式は検索開始位置を示す．N_2Sが含まれていたときはその位置が，含まれていなかったときは0が返される．

例）　X＝INSTR［1，0，〝 ″］

　　X＝INSTR（1，A$(0)，〝 ″）

❖使用法❖

各サブルーチンを使うためには，最初にアドレス・ポインタ領域，文字列領域，ワーク領域の初期設定をする必要があります．STRINITルーチンがこれを行ないます．このルーチン中の変数は，それぞれ次のものを示します．

ADRPOINT※	アドレス・ポインタ領域の先頭アドレス
STRTOP※	文字列領域の先頭アドレス
STREND	文字列領域の末尾アドレス
STRNUM	文字変数の個数
STRWORK※	ワーク領域の先頭アドレス

ユーザー側で設定するのは※印の3つで，現在はROMバージョンズで，スタック領域を前へ移動したものとして設定があります．文字列領域はアドレス・ポインタ領域のすぐ後ろにおき，ADRPOINTとSTRTOPとの間隔によって文字変数の個数が決まります．アドレス・ポインタは4バイトでワンセットなので，現在の設定では64個の文字変数が使えます．

上記に5つの変数をへたに変えると，暴走する恐れがあります．また，次の変数は各サブルーチンで使用するので注意してください．

STRPOINT, ADR, FLG, LNGTH, A, A1, B, I, J, L, N, S

文字列の長さは一応$FFバイトで押さえてありますが，11170行および22070行の$FFを適当な値に変えることきよって，もっと長くできます．この場合，ワーク領域は文字列の長さの上限＋1の広さが必要ですので，STRWORKの値を適宜変更してください．ただし，INPルーチンによるキーボードからの入力は，バッファ長の関係上$FFバイトまでしか受け付けません．

Kコンパイラは，引数として関数を持つことはできないようです（たとえばHEX［0，ASC［1］］など）．本当はできるのかも知れないが，やり方がまずいせいかことごとく暴走しました．そこで，ADD，CHR，STR，LEFT，MIDA，RIGHT，STRING，HEXの各サブルーチンは，結果を文字変数に代入する形式をとりました．このため，BASICでは1行ですむ簡単な関数が，大変手間のかかることもあります．

例）BASIC；

　　A$＝〝＊＊″＋MID$（B$，3，5)＝〝＊＊″

　K；

　　LET［0，〝＊＊″］：MIDA［2，1，3，5］；ADD［0，0，2］；ADD［0，0，〝＊＊″］

0番の文字変数がA$，1番がB$にあたりますが，余分な2番を使用したうえ，4段階に分ける必要があります．

CMP，POI，ASC，VAL，LEN，INSTRの各サブルーチンは関数ですが，それ以外のルーチンも一応代入先の文字変数（PRTルーチンでは出力する文字変数）のアドレスを返すようになっています．デバッグに役立ててください．

INPルーチンはモニタROMを呼び出しています．現在はFM-7用になっています．

なお，入力の中断は，BREAK，CTRL＋X，CTRL＋Cにて可能で，このときは空文字列が入力されます．しかし，暴走することがあるので避けた方がいいでしょう．

STRルーチンで，式の値が$8000のとき，返される文字列は〝－0″となりますが，これはKコンパイラの性格（数値が符号付き2バイト）からくるものです．修正しようかとも思いましたが，K自身が$8000を－0とプリントするので，合わせることにしました．

同じ理由でVALルーチンでも，符号付き2バイトをオー

バーフローするとき，変な値が返されることがあります．

引数の値は一部を除いてチェックしていませんので，ユーザー側で管理してください．また，引数が変数番号であるかアドレスであるかを見分けるのは，その引数が変数の個数の範囲内にあるかどうかをチェックしているだけなので，注意が必要です．

BASICにあって，このサブ・ルーチンにないものがありますが，次の理由により省きました．

OCT＄	めったに使わない
TIME＄	グラフィック・サブルーチン（I/O'83年4月号，FM-7/8活用研究）に掲載されている
SPACE＄	STRINGで代用可能

❖最後に❖

BASICでできることは，全部できるようにしようと欲張ったため，オン・メモリ版で裏RAMを使うか，ファイル版でなければコンパイルできないサイズになってしまいました．実際の使用にあたっては，必要な部分だけを入力すればよいのですが，別のサブルーチンが呼び出している部分もあるので気をつけてください．**STRINIT**と**LET**は最低限必要ですが，ほかに**LEFT**および**RIGHT**ルーチンは**MIDA**ルーチンを呼び出します．

2日で作り上げたものです，グラフィック・サブルーチンに次いで，Kの強力な機能を拡張できたと思います．変数の個数を極力押さえ，できるだけ多くをユーザーに開放しようとして，無茶苦茶な使い方をしているせいで，わかりにくいかもしれませんが，みなさん研究してみてください．

◘参考文献

〝FM-8活用研究〟，工学社

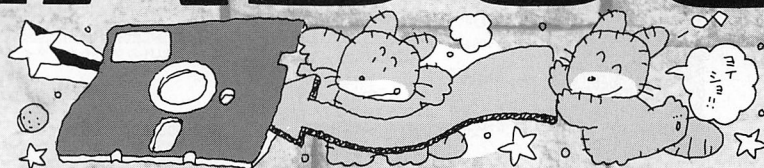
〝FM-7/8活用研究〟，工学社

Kコンパイラ文字変数ユーティリティソース・リスト

```
0 'GOSUB STRINIT
100 '
110 '
120 '        (*   USER'S  PROGRAM  AREA   *)
130 '
140 '
9990 'END
10000 '(* ████████ LET ████████ *)
10010 '
10020 'LET;
10030 '
10040 '   STRPOINT=ADRPOINT+%1*4
10050 '   A=%2
10060 '   ADR=STRPOINT(0)
10070 '   LNGTH=STRPOINT(1)
10080 '   IF A>=0 AND A<STRNUM THEN B=ADRPOINT+A*
4 ; A=B(0) ; L=B(1) ; GOTO LET1 ; FI
10090 '   L=0
10100 '   WHILE A:L)>0
10110 '   L=L+1
10120 '   WEND
10130 '   L=L+1
10140 'LET1;
10150 '   STRPOINT(1)=L
10160 '   I=2
10170 '   WHILE STRPOINT+I*2<STRTOP
10180 '   IF STRPOINT(I)<>0 THEN STRPOINT(I)=STRP
OINT(I)-LNGTH+L ; FI
10190 '   I=I+2
10200 '   WEND
10210 '   STREND=STREND-LNGTH+L
10220 '   IF L>LNGTH THEN GOTO LET2 ; FI
10230 '   I=0
10240 '   WHILE I<L
10250 '   ADR:I)=A:I)
10260 '   I=I+1
10270 '   WEND
10280 '   IF L=LNGTH THEN GOTO LET4 ; FI
10290 '   IF STRPOINT+4>=STRTOP THEN GOTO LET4 ;
FI
10300 '   FOR I=STRPOINT(2) TO STREND
10310 '   I:0)=I:LNGTH-L)
10320 '   NEXT
10330 '   GOTO LET4
10340 'LET2;
10350 '   IF STRPOINT+4>=STRTOP THEN GOTO LET3 ;
FI
10360 '   FOR I=STREND TO STRPOINT(2) STEP -1
10370 '   I:0)=I:LNGTH-L)
10380 '   NEXT
10390 'LET3;
10400 '   I=0
10410 '   WHILE I<L
10420 '   ADR:I)=A:I)
10430 '   I=I+1
10440 '   WEND
10450 'LET4;
10460 '   A=ADR
10470 '   RETURN
10480 '
11000 '(* ████████ ADDITION ████████ *)
11010 '
11020 'ADD;
11030 '
11040 '   ADR=%1
11050 '   A=%2
11060 '   A1=%3
11070 '   IF A>=0 AND A<STRNUM THEN B=ADRPOINT+A*
4 ; A=B(0) ; FI
11080 '   IF A1>=0 AND A1<STRNUM THEN B=ADRPOINT+
A1*4 ; A1=B(0) ; FI
11090 '   J=STRWORK
11100 '   I=0
11110 '   WHILE A:I)>0
11120 '   B=A:I)
11130 '   CPOKE J,B
11140 '   I=I+1
11150 '   WEND
11160 '   I=0
11170 '   WHILE A1:I)>0 AND J-STRWORK<$FF
11180 '   B=A1:I)
11190 '   CPOKE J,B
11200 '   I=I+1
11210 '   WEND
11220 '   CPOKE J,0
11230 '   LET [ADR,STRWORK]
11240 '   RETURN
11250 '
12000 '(* ████████ COMPARE ████████ *)
12010 '
12020 'CMP;
12030 '
12040 '   A=%1
12050 '   A1=%2
12060 '   IF A>=0 AND A<STRNUM THEN B=ADRPOINT+A*
4 ; A=B(0) ; FI
12070 '   IF A1>=0 AND A1<STRNUM THEN B=ADRPOINT+
A1*4 ; A1=B(0) ; FI
12080 '   FLG=0
12090 '   I=0
12100 '   WHILE A:I)=A1:I) AND A:I)>0
12110 '   I=I+1
12120 '   WEND
12130 '   IF A:I)+A1:I)>0 THEN FLG=SGN(A:I)-A1:I)
) ; FI
12140 '   B=FLG
12150 '   RETURN
12160 '
13000 '(* ████████ INPUT ████████ *)
13010 '
13020 'INP;
13030 '
```

```
13040 '   ADR=%1
13050 '   CODE $34,$7F,$4F,$1F,$8B,$BD,$D8,$07,$3
5,$7F
13060 '   A=$43D
13070 '   IF A:0)=1 THEN A:0)=0 ; FI
13080 '   I=0
13090 '   WHILE A:I)>0
13100 '   STRWORK:I)=A:I)
13110 '   I=I+1
13120 '   WEND
13130 '   STRWORK:I)=0
13140 '   LET [ADR,STRWORK]
13150 '   RETURN
13160 '
14000 '(* ■■■■■■■■ PRINT ■■■■■■■■ *)
14010 '
14020 'PRT;
14030 '
14040 '   A=%1
14050 '   B=ADRPOINT+A*4
14060 '   ADR=B(0)
14070 '   LNGTH=B(1)
14080 '   FOR I=0 TO LNGTH-2
14090 '   ? CHR$(ADR:I))
14100 '   NEXT
14110 '   B=ADR
14120 '   RETURN
14130 '
15000 '(* ■■■■■■■■ POINTER ■■■■■■■■ *)
15010 '
15020 'POI;
15030 '
15040 '   A=%1
15050 '   B=ADRPOINT+A*4
15060 '   ADR=B(0)
15070 '   RETURN
15080 '
16000 '(* ■■■■■■■■ CHR$ ■■■■■■■■ *)
16010 '
16020 'CHR;
16030 '
16040 '   ADR=%1
16050 '   STRWORK(0)=%2*256
16060 '   LET [ADR,STRWORK]
16070 '   RETURN
16080 '
17000 '(* ■■■■■■■■ STR$ ■■■■■■■■ *)
17010 '
17020 'STR;
17030 '
17040 '   ADR=%1
17050 '   B=%2
17060 '   S=ABS(B)
17070 '   J=STRWORK
17080 '   IF B<0 THEN B='-' ELSE B=' ' ; FI
17090 '   CPOKE J,B
17100 '   I=10000
17110 '   FLG=0
17120 '   WHILE I>0
17130 '   N=S/I
17140 '   IF N>0 OR FLG=1 THEN FLG=1 ; S=S-N*I ;
N=N+$30 ; CPOKE J,N ; FI
17150 '   I=I/10
17160 '   WEND
17170 '   IF FLG=0 THEN CPOKE J,$30 ; FI
17180 '   CPOKE J,0
17190 '   LET [ADR,STRWORK]
17200 '   RETURN
17210 '
18000 '(* ■■■■■■■■ LEFT$ ■■■■■■■■ *)
18010 '
18020 'LEFT;
18030 '
18040 '    MIDA [%1,%2,1,%3]
18050 '    RETURN
18060 '
19000 '(* ■■■■■■■■ MID$ A ■■■■■■■■ *)
19010 '
19020 'MIDA;
19030 '
19040 '   ADR=%1
19050 '   A=%2
19060 '   I=%3-1
19070 '   J=%4-1
19080 '   IF I<0 THEN GOTO MIDA2 ; FI
19090 '   IF A>=0 AND A<STRNUM THEN B=ADRPOINT+A*
4 ; A=B(0) ; L=B(1) ; GOTO MIDA1 ; FI
19100 '   L=0
19110 '   WHILE A:L)>0
19120 '   L=L+1
19130 '   WEND
19140 '   L=L+1
19150 'MIDA1;
19160 '   IF I>=L-1 THEN LET [ADR,""] ; GOTO MIDA
2 ; FI
19170 '   S=STRWORK
19180 '   FOR N=I TO I+J
19190 '   B=A:N)
19200 '   CPOKE S,B
19210 '   NEXT
19220 '   CPOKE S,0
19230 '   LET [ADR,STRWORK]
19240 'MIDA2;
19250 '   RETURN
19260 '
20000 '(* ■■■■■■■■ MID$ B ■■■■■■■■ *)
20010 '
20020 'MIDB;
20030 '
20040 '   ADR=%1
20050 '   I=%2-1
20060 '   J=%3
20070 '   A=%4
20080 '   B=ADRPOINT+ADR*4
20090 '   ADR=B(0)
20100 '   IF I<0 THEN GOTO MIDB1 ; FI
20110 '   IF A>=0 AND A<STRNUM THEN B=ADRPOINT+A*
4 ; A=B(0) ; FI
20120 '   N=0
20130 '   WHILE ADR:N)>0 AND N<I
20140 '   N=N+1
20150 '   WEND
20160 '   IF ADR:N)=0 THEN GOTO MIDB1 ; FI
20170 '   WHILE ( ADR:N)>0 AND A:N-I)>0 ) AND N-I
<J
20180 '   ADR:N)=A:N-I)
20190 '   N=N+1
20200 '   WEND
20210 'MIDB1;
20220 '   B=ADR
20230 '   RETURN
20240 '
21000 '(* ■■■■■■■■ RIGHT$ ■■■■■■■■ *)
21010 '
21020 'RIGHT;
21030 '
21040 '   ADR=%1
21050 '   A=%2
21060 '   I=%3
21070 '   IF A>=0 AND A<STRNUM THEN B=ADRPOINT+A*
4 ; A=B(0) ; L=B(1)-1 ; GOTO RIGHT1 ; FI
21080 '   L=0
21090 '   WHILE A:L)>0
21100 '   L=L+1
21110 '   WEND
21120 'RIGHT1;
21130 '   B=L-I+1
21140 '   IF B<=0 THEN B=1 ; FI
21150 '   MIDA [ADR,A,B,I]
21160 '   RETURN
21170 '
22000 '(* ■■■■■■■■ STRING$ ■■■■■■■■ *)
22010 '
22020 'STRING;
22030 '
22040 '   ADR=%1
22050 '   L=%2
22060 '   B=%3
22070 '   IF L>$FF THEN L=$FF ELSE IF L<0 THEN L=
0 ; FI ; FI
22080 '   FOR I=0 TO L-1
22090 '   STRWORK:I)=B
22100 '   NEXT
22110 '   STRWORK:L)=0
22120 '   LET [ADR,STRWORK]
22130 '   RETURN
22140 '
23000 '(* ■■■■■■■■ HEX$ ■■■■■■■■ *)
23010 '
23020 'HEX;
23030 '
23040 '   ADR=%1
23050 '   B=%2
23060 '   J=STRWORK
```

Kコンパイラ文字変数ユーティリティソース・リスト

```
23070 '   S=HIGH(B)
23080 '   GOSUB HEX1
23090 '   S=LOW(B)
23100 '   GOSUB HEX1
23110 '   CPOKE J,0
23120 '   LET [ADR,STRWORK]
23130 '   RETURN
23140 '
23150 'HEX1;
23160 '   I=$10
23170 '   WHILE I>0
23180 '   N=S/I
23190 '   S=S-N*I
23200 '   IF N<10 THEN N=N+$30 ELSE N=N+$37 ; FI
23210 '   CPOKE J,N
23220 '   I=I/16
23230 '   WEND
23240 '   RETURN
23250 '
24000 '(* ■■■■■■■■■■ ASC ■■■■■■■■■■ *)
24010 '
24020 'ASC;
24030 '
24040 '   A=%1
24050 '   IF A>=0 AND A<STRNUM THEN B=ADRPOINT+A*
4 ; A=B(0) ; FI
24060 '   N=A:0)
24070 '   RETURN
24080 '
25000 '(* ■■■■■■■■■■ VAL ■■■■■■■■■■ *)
25010 '
25020 'VAL;
25030 '
25040 '   A=%1
25050 '   IF A>=0 AND A<STRNUM THEN B=ADRPOINT+A*
4 ; A=B(0) ; FI
25060 '   FLG=0
25070 '   N=1
25080 '   S=0
25090 '   I=-1
25100 '   REPEAT
25110 '   I=I+1
25120 '   UNTIL A:I)<>' '
25130 '   IF A:I)='$' THEN I=I+1 ; GOTO VAL2 ; FI
25140 '   IF A:I)='-' THEN N=-1 ; I=I+1 ; FI
25150 '   WHILE FLG=0
25160 '   J=A:I)
25170 '   IF J=0 THEN FLG=1 ; GOTO VAL1 ; FI
25180 '   IF J=' ' THEN GOTO VAL1 ; FI
25190 '   IF J<'0' OR J>'9' THEN FLG=1 ; GOTO VAL
1 ; FI
25200 '   S=S*10+J-$30
25210 'VAL1;
25220 '   I=I+1
25230 '   WEND
25240 '   S=S*N
25250 '   RETURN
25260 '
25270 'VAL2;
25280 '   WHILE FLG=0
25290 '   J=A:I)
25300 '   IF J=0 THEN FLG=1 ; GOTO VAL4 ; FI
25310 '  IF J=' ' THEN GOTO VAL4 ; FI
25320 '   IF J>='0' AND J<='9' THEN J=J-$30 ; GOT
O VAL3 ; FI
25330 '   IF J>='A' AND J<='F' THEN J=J-$37 ; GOT
O VAL3 ; FI
25340 '   FLG=1 ; GOTO VAL4
25350 'VAL3;
25360 '   S=S*16+J
25370 'VAL4;
25380 '   I=I+1
25390 '   WEND
25400 '   S=S*N
25410 '   RETURN
25420 '
26000 '(* ■■■■■■■■■■ LEN ■■■■■■■■■■ *)
26010 '
26020 'LEN;
26030 '
26040 '   A=%1
26050 '   IF A>=0 AND A<STRNUM THEN B=ADRPOINT+A*
4 ; L=B(1)-1 ; GOTO LEN1 ; FI
26060 '   L=0
26070 '   WHILE A:L)>0
26080 '   L=L+1
26090 '   WEND
26100 'LEN1;
26110 '   B=L
26120 '   RETURN
26130 '
27000 '(* ■■■■■■■■■■ INSTR ■■■■■■■■■■ *)
27010 '
27020 'INSTR;
27030 '
27040 '   S=%1-1
27050 '   A=%2
27060 '   A1=%3
27070 '   IF S<0 THEN S=0 ; GOTO INSTR1 ; FI
27080 '   IF A>=0 AND A<STRNUM THEN B=ADRPOINT+A*
4 ; A=B(0) ; FI
27090 '   IF A1>=0 AND A1<STRNUM THEN B=ADRPOINT+
A1*4 ; A1=B(0) ; FI
27100 '   I=-1
27110 '   FLG=0
27120 '   WHILE FLG=0
27130 '   IF I=S THEN FLG=1 ; FI
27140 '   IF A:I)=0 THEN FLG=2 ; FI
27150 '   I=I+1
27160 '   WEND
27170 '   IF FLG=2 THEN S=0 ; GOTO INSTR1 ; FI
27180 '   FLG=0
27190 '   WHILE FLG=0 AND A:S)>0
27200 '   GOSUB INSTR2
27210 '   S=S+1
27220 '   WEND
27230 '   IF FLG=0 THEN S=0 ; FI
27240 'INSTR1;
27250 '   B=S
27260 '   RETURN
27270 '
27280 'INSTR2;
27290 '   I=0
27300 '   FLG=1
27310 '   WHILE A1:I)>0 AND FLG=1
27320 '   IF A:I+S)<>A1:I) THEN FLG=0 ; FI
27330 '   I=I+1
27340 '   WEND
27350 '   RETURN
27360 '
28000 '(* ■■■■■■■■■■ STRING INITIALIZE ■■■■■■■■■■
■ *)
28010 '
28020 'STRINIT;
28030 '
28040 '   ADRPOINT=$7000
28050 '   STRTOP=$7100
28060 '   STRWORK=$7F00
28070 '   STRNUM=(STRTOP-ADRPOINT)/4
28080 '   STREND=STRTOP+STRNUM-1
28090 '   FOR I=0 TO STRNUM*2 STEP 2
28100 '   ADRPOINT(I)=STRTOP+I/2
28110 '   ADRPOINT(I+1)=1
28120 '   NEXT
28130 '   FOR I=STRTOP TO STREND
28140 '   I:0)=0
28150 '   NEXT
28160 '   RETURN
```

DISK BASICでは，ROM BASIC用ソフトが実行できないなどの不満をお持ちのFM-7ユーザーのために，ROM BASICなみのフリーエリアを誇るURADOSを紹介します．

★URADOSの特徴★

❶ROM BASICなみのフリーエリア（マシン語は$7FFFまでOK）．
❷マシン語プログラムなどの開発に威力を発揮するスーパーモニタを利用できます．しかも裏RAM上にLOADされるため，ユーザーエリアを浸食しません．
❸プログラムのセーブ時間の短縮．
❹RS-232CからASCII型式のプログラムがロードできます（**LOAD"COM0:"**）．

★DISK BASICとの互換性★

URADOS上でDISK BASICのディスケットが読み書きでき，その逆もまた可能です．つまり，いま使っているディスケットはそのまま利用できます．また，ランダム・ファイル関係の命令は利用できませんが，シーケンシャル・ファイルならば，無条件に15個までOPENできます．

★スーパーモニタについて★

URADOSにはマシン語の入力などに便利なスーパーモニタが付属しています．コマンドは，**H**（ヘルプ）ですべて表示されますが，その概略を以下に示します．

B：ブレーク・ポイントの設定（最大10個）
Gコマンド時のみ有効で，指定アドレスを**SWI**に置き換えて実行します．ですから，ユーザーは**SWI**を利用できません．
C：サブルーチン・コール
D：メモリ・ダンプ
80桁モード時は，ASCIIダンプも出力します．
E：メモリ・チェンジ
80桁モード時のみ利用でき，強制的に単色モードになります．このコマンドは，多くのサブ･コマンドを持っており，**H**を押すことにより，利用法が表示されます．主な特徴は，縦横チェックサム，16進テン･キー，スクリーン･エディットなどです．
F：メモリ初期化
指定範囲を定数（文字定数可）で初期化します．
G：実行
アドレス指定を省略した場合は，仮想レジスタ（PC）の内容で指定されるアドレスにジャンプします．
H：ヘルプ
80桁モード時には，コマンドの機能も表示します．
L：ディスアセンブル
M：ブロック転送
P：プリンタ・トグル
D，L，Xコマンド時に，プリンタに出力するかどうかを指定します．
Q：BASIC復帰
CTRL＋C，CTRL＋X，BREAKキーでも同様な動作をします．
R：リード・ディスク
クラスタNoによる指定も可．
S：サーチ・コンスタント
指定範囲から最大16要素までの定数列をさがしだします．
W：ライト・ディスク
X：レジスタ（仮想レジスタ）の表示・変更
D，Lコマンドで表示された16進ダンプの部分は，一度に複数行に渡ってスクリーン・エディットできます．
$8000以降は，ROMおよびシステム領域ですから，書き込みはできません．
スーパーモニタは別ファイルになるので，URADOSをBOOT後，以下のステートメントを実行することにより利用できます．

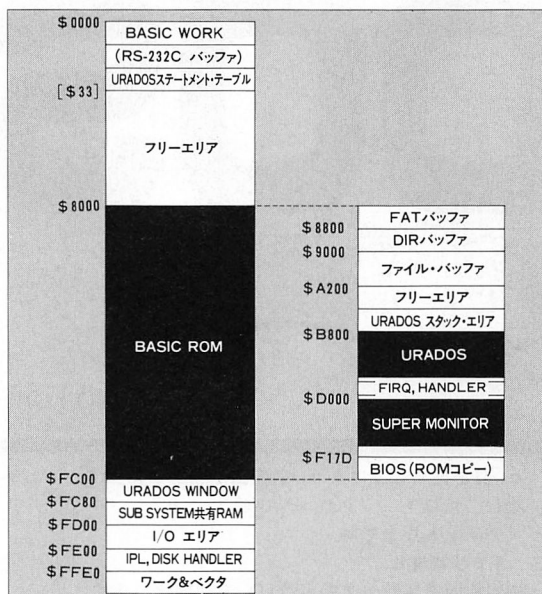
```
CLEAR，&H5000:LOADM"MONITOR",,R
```

★プログラムの解説★

●リスト1
URADOSのIPL，ブートローダー，本体およびSYSGENプログラムです．フォーマット済のディスケットをドライブ0に入れて，**EXEC&H3000**を実行すると，ディスケットにシステムを書き込みます．このとき，エラーが起きると，

図1　メモリ・マップ

アドレス	内容
$0000	BASIC WORK
	(RS-232C バッファ)
	URADOSステートメント・テーブル
[$33]	フリーエリア
$8000	BASIC ROM
$FC00	URADOS WINDOW
$FC80	SUB SYSTEM共有RAM
$FD00	I/O エリア
$FE00	IPL, DISK HANDLER
$FFE0	ワーク&ベクタ

アドレス	内容（$8000〜$FC00）
$8000	FATバッファ
$8800	DIRバッファ
$9000	ファイル・バッファ
$A200	フリーエリア
	URADOS スタック・エリア
$B800	URADOS
	FIRQ, HANDLER
$D000	SUPER MONITOR
$F17D	BIOS(ROMコピー)

"Abort"を表示して実行を中断します．終わったら，**DISKINI0**を実行しておきましょう．

●リスト 2

スーパーモニタ本体と，ローダーのプログラムです．打ち込み終わったら，**リスト1**で作成したURADOSのディスケットに**SAVEM"MONITOR", &H5000, &H6BFF, &H5000**で，セーブしておきます．

●リスト 3

スキュー付きディスク・フォーマット・プログラムです．このプログラムでフォーマットしたディスケットを使用すると，LOAD，SAVEの時間が2.5倍程度向上します．ただし，DISK BASICの"VOLCOPY"を利用した場合，逆に3.7倍程度時間がかかります．このフォーマット・プログラムは，DISK BASICのディスケットに利用しても効果があります．

●リスト 4

スタート・アップ・プログラムの例と，フリーエリア拡張プログラムです．URADOSは，ブート時にディレクトリを調べ，"Startup"というファイルが存在すると，そのプログラムを自動的に実行します．２つ目のプログラムは，URADOSでROM BASICとまったく同じフリーエリアを利用できるようにするプログラムです．ただし，このプログラムを利用すると，次のような制限がつくので，大きなゲーム・プログラムを実行する場合以外は，利用しない方がよいでしょう．

❶次のステートメントが利用できなくなる．

DSKINI, NAME, DSKF

❷通常のURADOSでは，DISK BASIC用のプログラムを修正できるように，**CVI, MKI$**などのステートメント・テーブルを持っていますが，フリーエリアを拡張するとこのテーブルがなくなり，DISK BASIC用のプログラムを修正してURADOSで利用するといったことが困難になります（たとえば，リストを取っても**CVI, DSKI$**などが表示されない）．

このプログラムは市販のURADOSにも利用できます．

図2　FILESの表示

"ファイル名"：'タイプ　モード　ユーズド・クラスタ

ファイル型式	ファイル・タイプ	ファイル・モード
ASCIIセーブ・テキスト	BAS	ASC
中間コード・セーブ・テキスト	BAS	BIN
ランダム・ファイル	DAT	RDM
シーケンシャル・ファイル	DAT	SEQ
マシン語・ファイル	OBJ	

上記以外のファイル・タイプは，DISK BASICと同じ表示になります．

表1　使用ROMエントリ・アドレス

```
$00DE   BIOS Call
$8597   block transfer
$863E   hot start with F-BASIC title
$8DD1   error entry
$8E35   ROM BASIC error trap
$8E69   hot start after error
$8F39   NEW
$8F91   clear all variable
$8FE1   Break entry ( hot start with stack recovery )
$9288   parenthesis check
$974C   return 1 byte result
$974D   return 2 byte result
$99BC   get 1 byte parameter with formula processing
$99BF   get 1 byte parameter from FAC
$9B47   CR LF
$C28C   translate text into intermediate code
$C65D   ON ERROR trap
$C67A   interpret
$CC37   set file descripter
$C8BC   put console parameter into sub system
$CE4B   close file ( buffer #($BF) )
$CE81   close all file
$CEFC   open file
$CF9C   LOAD routine body
$D0E5   set POS result
$D678   check abort
$D680   abort entry
$D93D   field init and erase end of line
$DAEF   cursor off
$DAF6   cursor on
$EC05   P.S.G. sound stop
$FBFE   BASIC cold start vector
$FE02   restore disk
$FE05   write disk
$FE08   read disk
```

また，スーパーモニタは，自動的にロードされるので，スタートアップ・ファイルは不要です．

★URADOSの内部構造★

表1にURADOSの利用しているROMエントリ・アドレス，**図1**にメモリ・マップを示します．

❶RS-232Cバッファについて

RS-232Cカードが実装されている場合は，自動的に領域を確保するので，ゲームを走らせる場合など，フリーエリアが不足する場合は，カードを抜いてブートしましょう．

❷URADOSステートメント・テーブル（リスト4解説参照）

❸URADOS WINDOWについて

ROMと裏RAMの連絡や，ブレイク・ポイントの実行などのルーチンがあり，URADOSのルーチンを利用する場合は主にここを通してCALLされます．また，この領域は，IPL ROMのスタック領域になっているので，ストップ・リセットなどを実行すると破壊されます．

❹BIOS（ROMコピー）について

裏RAM上でもBIOSが利用できるようにROMのBIOSがコピーしてあります．

★URADOSを利用するにあたって★

ストップ・リセットによる中断は極力避けてください（前

項の❸参照). もし，中断した場合はすぐにEXEC&HFC00を実行すると復活する場合があります. 裏RAMや$FC00～$FC7Fはユーザーは利用できません.

プログラム・ファイルのセーブ時には，最後に1回FATを書きかえるのみなので，セーブを途中で中断した場合，結果はまったく残りません.

FILESの出力結果を使ってロード，セーブができます. 図2に表示の見方を示します.

エラーコード74 (DISK BASIC Feature) が新設され，ランダム・ファイル関係の命令を利用しようとするとこのエラーが発生します.

★おわりに★

強力なモニタを搭載したURADOSは，きっとお役にたつものと思います. また，機会があれば，URADOS用のファイル転送ユーティリティなどを発表したいと思います.

リスト1　SYSGENプログラム　ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3000 1A 50 86 FD 1F 8B 8E 30 6E C6 01 8D 12 8E 30 76 :5D
3010 C6 03 8D 0B 8E 30 7E C6 10 8D 04 4F 1F 8B 39 34 :6A
3020 04 8D 1B CC 01 00 E3 02 ED 02 6C 05 A6 05 81 11 :FB
3030 25 06 86 01 A7 05 6C 06 35 04 5A 26 E2 39 C6 05 :6F
3040 34 04 34 10 8E 30 66 BD FE 02 35 10 81 0A 27 0F :63
3050 BD FE 05 35 04 4D 27 0D 81 0A 27 03 5A 26 E1 4F :DF
3060 1F 8B 7E D6 80 39 08 E5 E5 E5 E5 E5 E5 00 09 00 :26
3070 38 00 00 01 00 00 09 00 3C F0 00 03 00 00 09 00 :7A
3080 40 00 00 01 01 00 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :34
3090 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
30A0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
30B0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
30C0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
30D0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
30E0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
30F0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
---------------------------------------------------------
Sum: D4 B6 AE 35 AB B9 21 D5 68 62 34 2A A1 AF F2 46 :77
```

注) 3100～37FFまでは00としてください.

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3800 20 2B 54 68 69 73 20 69 73 20 49 50 4C 20 70 72 :E6
3810 6F 67 72 61 6D 20 6F 66 20 55 52 41 44 4F 53 20 :19
3820 49 2F 4F 20 56 65 72 73 69 6F 6E 2E 00 1A 50 10 :75
3830 CE 01 00 86 FD 1F 8B 8E 01 A7 C6 03 8D 19 B6 4C :A3
3840 F0 81 53 26 52 8E 01 AF C6 10 8D 0B 4F 1F 8B 8E :6F
3850 4D 00 43 6E 9F FB FE 34 04 8D 1B EC 02 C3 01 00 :28
3860 ED 02 6C 05 A6 05 81 11 25 06 86 01 A7 05 6C 06 :6D
3870 35 04 5A 26 E2 39 C6 05 34 04 BD FE 08 35 04 4D :20
3880 27 1C 81 0A 27 11 34 14 8E 01 9F BD FE 02 35 14 :82
3890 81 0A 27 03 5A 26 E1 4F 1F 8B 6E 9F FB FE 39 08 :56
38A0 39 08 E5 E5 E5 E5 00 0A 00 4C F0 00 03 00 00 0A :28
38B0 00 50 00 00 01 01 00 01 00 E5 E5 E5 E5 E5 E5 E5 :96
38C0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
38D0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
38E0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
38F0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
---------------------------------------------------------
Sum: 7A 5B 92 B4 9D 8F 7B CB 61 83 30 8D 92 37 AC 6E :11
```

注) 3900～3CEFまでは00としてください.

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3CF0 53 00 00 00 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :0F
3D00 34 40 CE 80 00 10 8E 7C 00 7F FD 0F 6F C0 31 3F :06
3D10 26 FA 8E 50 00 CE B8 00 10 8E 10 00 A6 84 6F 80 :4B
3D20 A7 C0 31 3F 26 F6 86 39 B7 D0 00 7D FD 0F BE FB :7B
3D30 FA A6 84 7F FD 0F A7 80 7D FD 0F 8C FC 00 26 F1 :FE
3D40 86 19 B7 03 0D 4A B7 03 0F 86 50 97 C3 BD C8 BC :EA
3D50 8E 4E 8A BD 9B DB BD DB 49 35 40 31 C4 8E 4E 2E :EE
3D60 C6 5D BD 85 97 33 41 DF 33 10 BF 01 FA 31 A8 1C :41
3D70 10 BF 01 FF 31 A8 21 10 BF 01 FC 31 A8 10 10 BF :4D
3D80 02 01 86 7E 7F FD 0F FE B8 09 7D FD 0F 8E 02 2D :97
3D90 C6 0C A7 80 EF 81 33 42 5A 26 F7 B7 02 6F FF 02 :7E
3DA0 70 33 42 B7 02 69 FF 02 6A 33 42 C6 80 F7 01 EB :10
3DB0 8E 7F FF BF 05 9D 9F 45 30 89 FE D4 9F 3F 1F 14 :ED
3DC0 8E 4E CD BD 86 CF BD 8F 39 8E 87 58 9F D9 03 47 :6F
3DD0 03 48 86 01 B7 05 E5 BD 9B 50 7F FD 0F BD B8 00 :1B
3DE0 8E 4E 24 CE 02 DE C6 08 A6 80 A7 C0 5A 26 F9 BD :3F
---------------------------------------------------------
Sum: 1D C6 F5 D2 2C FE 76 C2 99 D4 AD 5A 54 B3 0C 87 :1A
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3DF0 B8 14 6D 9F B8 0C 27 21 6D 4B 26 1D EC 4C 26 19 :56
3E00 7D FD 0F 8E 4E 1F CE 04 3D DF D9 C6 10 BD 85 97 :FA
3E10 BD C2 8C BD DA EF BD C6 7A 7D FD 0F 7E 86 3E 52 :AB
3E20 55 4E 20 22 73 74 61 72 74 75 70 20 22 00 44 53 :D1
3E30 4B 49 4E C9 44 53 4B 4F A4 4E 41 4D C5 46 49 4C :FC
3E40 45 C4 4C 53 45 D4 52 53 45 D4 44 53 4B C6 43 56 :C0
3E50 C9 43 56 D3 43 56 C4 4D 4B 49 A4 4D 4B 53 A4 4D :F3
3E60 4B 44 A4 4C 4F C3 44 53 4B 49 A4 81 EE 25 04 6E :66
3E70 9F 02 06 1A 50 7F FD 0F 7E B8 03 C1 68 25 04 6E :95
3E80 9F 02 0B 1A 50 7F FD 0F 7E B8 06 55 52 41 20 44 :29
3E90 4F 53 20 66 6F 72 20 46 4D 2D 37 20 20 5B 20 49 :24
3EA0 2F 4F 20 56 65 72 73 69 6F 6E 20 5D 0D 0A 43 6F :CA
3EB0 70 79 72 69 67 68 74 28 43 29 31 39 38 33 20 42 :D2
3EC0 79 20 44 2E 4F 2E 53 2E 0D 0A 1B 67 00 05 41 55 :3D
3ED0 54 4F 20 05 4C 49 53 54 0D 04 52 55 4E 0D 06 46 :63
3EE0 49 4C 45 53 0D 0A 46 49 4C 45 53 22 31 3A 22 0D :73
---------------------------------------------------------
Sum: 2D 8F 28 26 F1 99 A5 5F 78 57 8A 2A 83 5D 71 06 :72
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3EF0 05 53 41 56 45 22 06 4C 4C 49 53 54 0D 09 52 55 :A1
3F00 4E 20 20 22 31 3A 0D 05 4C 4F 41 44 0D 09 4C 4F :FE
3F10 41 44 20 22 31 3A 0D 00 50 72 6F 67 72 61 6D 65 :7C
3F20 64 20 62 79 20 54 6F 6D 2E 57 61 74 61 6E 61 62 :9B
3F30 65 00 42 6F 6F 74 20 6C 6F 61 64 65 72 20 65 6E :83
3F40 64 2E 00 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :33
3F50 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3F60 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3F70 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3F80 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3F90 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3FA0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3FB0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3FC0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3FD0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3FE0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
---------------------------------------------------------
Sum: B3 F7 17 59 0D 35 86 01 5C 99 9F AF 36 D8 A8 B0 :8C
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4000 7E B9 15 7E B9 9E 7E B9 B9 FC 3F 01 C5 B2 E5 E5 :8E
4010 E5 7E B8 D4 7E BC 5F 7E B8 2D 7E B8 3E 7E B8 4A :DF
4020 7E B8 87 7E BA 22 7E FC 1A 7E FC 20 00 34 10 8E :17
4030 B8 53 9D DE B6 B8 60 27 F6 B6 B8 5F 35 90 34 10 :47
4040 8E B8 59 B7 B8 5F 9D DE 35 90 7D 05 AC 27 EF 8D :7E
4050 10 20 EB 15 00 B8 5F E5 E5 14 00 B8 5F 00 01 E5 :22
4060 E5 34 10 8E B8 7E 9D DE E6 01 10 26 02 8E B7 B8 :84
4070 5F 8E B8 80 9D DE E6 01 10 26 02 80 35 90 17 00 :1B
4080 0E 00 B8 5F 00 01 39 34 10 10 8E 00 00 34 10 6D :F2
4090 80 27 04 31 21 20 F8 8E B8 CC 10 AF 04 35 20 27 :66
40A0 E5 10 AF 02 F6 05 AC 27 1B 34 10 8E B8 7E 9D DE :12
40B0 35 10 E6 01 10 26 02 44 86 0E A7 84 9D DE E6 01 :C9
40C0 10 26 02 38 86 14 A7 84 9D DE 35 90 E5 E5 E5 E5 :09
40D0 E5 E5 E5 E5 C1 10 25 01 39 10 8E B8 F5 58 10 AE :25
40E0 A5 33 62 10 FF B8 0E 10 CE B7 F0 F6 B8 10 AD A4 :A3
40F0 10 FE B8 0E 39 B9 15 B9 67 C2 B3 BD FF BD FF BB :A3
---------------------------------------------------------
Sum: CD 5F 4F 56 5A 88 08 77 05 AD BB 57 64 08 F3 5C :B1
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4100 FA BD BB BF A5 C0 0A C0 71 C1 2C C2 38 C1 50 C4 :8D
4110 27 D0 00 C4 84 8E C5 B8 CE FC 00 C6 80 8D 1C 86 :89
4120 80 B7 02 E6 7F 80 00 7F 80 9E 7F 81 3C 7F 81 DA :D1
4130 7F B8 2C 8E B9 43 FE 01 E1 C6 24 A6 80 A7 C0 5A :9E
4140 26 F9 39 34 06 B6 FD 04 85 02 26 12 C6 40 F7 FD :02
4150 05 F7 03 13 7F FD 05 F6 FD 04 C5 02 27 F9 A6 62 :79
4160 8A 40 A7 62 35 06 3B 32 C4 81 04 10 26 01 91 CC :58
4170 00 FF FD 02 DB 10 8E CE 81 BD FC 20 C6 11 D7 BF :0C
4180 86 10 10 8E CE FC BD FC 20 BE 02 B2 AE 08 AE 88 :35
4190 11 86 26 A7 01 10 8E CF 9C 34 20 7E FC 1A 8E B9 :9D
41A0 D9 80 E8 48 1F 89 10 FF B8 0E 10 CE B7 F0 8D 45 :5D
41B0 AD 95 10 FE B8 0E 7E FC 1A C0 56 10 FF B8 0E 10 :A5
41C0 CE B7 F0 C1 0E 10 22 04 36 34 04 10 8E 92 88 BD :5D
41D0 FC 20 35 04 8E B9 E5 6E 95 BB 55 BD FF BD 43 BD :0D
41E0 FF BD FF BD FF C2 08 BD FF BD FF BD FF BD FF BD :EE
41F0 FF BD FF BD FF 0C DA 26 02 0C D9 A6 9F 00 D9 81 :09
---------------------------------------------------------
Sum: BA 27 1A 5C 36 14 5A 0D C1 DD 73 31 D8 95 2C B6 :99
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4200 3A 24 08 81 20 27 EE 80 30 80 D0 39 34 10 8E BA :E1
4210 1A B6 01 E5 A7 07 9D DE 35 90 14 00 BA 20 00 02 :94
4220 11 00 8D E8 7E B8 87 34 10 8E BA 31 BD BA 22 35 :CE
4230 90 0D 0A 00 10 8E D6 80 10 FE B8 0E 34 20 7E FC :3D
4240 1A 34 30 10 8E 9B 47 BD FC 20 8D D6 BD B8 2D BE :9A
4250 03 12 26 E0 84 DF 81 59 27 05 81 4E 26 EE 4D 34 :E8
4260 01 BD B8 4A BD BA 27 35 B1 96 47 4C 27 01 39 8E :5C
4270 BA 7D 8D CD 27 F8 10 8E 8F E1 7E BA 38 41 72 65 :46
4280 20 79 6F 75 20 73 75 72 65 20 28 20 59 20 2F 20 :8C
4290 4E 20 29 20 3F 20 00 34 06 F6 C5 AF 8C 34 06 6F :EF
42A0 E2 34 04 86 20 3D E3 E1 8E 90 00 30 8B 6D 84 35 :C0
42B0 86 34 06 B6 BB 47 C6 9E 3D 8E 80 00 30 8B 6D 84 :D3
42C0 35 86 34 7C 86 FD 1F 8B 86 05 34 02 8E BB 09 F6 :A1
42D0 BB 49 58 AD 95 4D 27 1A C6 48 81 0A 27 14 C6 49 :0F
42E0 81 0B 27 0E C6 35 6A E4 27 08 8D 25 C6 48 81 0A :84
42F0 26 DA 32 61 4D 35 7A 26 09 1F 89 39 C6 35 20 02 :BC
---------------------------------------------------------
Sum: 3A 1C C2 BE B3 6B 2F BF 9A E0 61 0B 02 8A E9 65 :A2
```

リスト1　SYSGENプログラム　ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4300 C6 05 10 8E 8D D1 7E BA 38 BB 11 BA FB BB 1B BB :49
4310 1E 8E BB 40 86 08 A7 84 7E FE 02 86 0A 8C 86 09 :89
4320 8E BB 40 A7 84 B6 BB 48 6F 06 81 10 25 04 6C 06 :0E
4330 80 10 4C A7 05 86 0A A1 84 10 27 42 CB 7E FE 05 :02
4340 E5 E5 E5 E5 00 01 00 00 00 E5 BD B9 FB 26 01 39 :4B
4350 C6 02 7E BB 02 10 8E 99 BC BD FC 20 8D EC F1 B8 :F1
4360 0B 2F 05 C6 3C 7E BB 02 F7 BB 47 BD BA 69 C6 0F :2A
4370 BD BA 9D 27 21 A6 01 B1 BB 47 26 1A 6F 84 34 04 :21
4380 CE 02 B4 C6 12 A6 C0 2A 08 84 0F A1 E4 26 02 6F :A3
4390 5F 5A 26 F1 35 04 5A 2A D7 BD BA B1 6F 84 8E 90 :9D
43A0 0E 6F 01 33 02 30 01 BF BB 42 C6 01 F7 BB 44 8D :EA
43B0 29 F7 BB 48 BD BA C2 86 FF A7 84 34 10 30 05 A6 :2B
43C0 80 81 FE 26 05 4C A7 1F 20 F5 35 10 C6 03 F7 BB :11
43D0 48 BD BA C2 5C C1 20 25 F5 39 86 02 B7 BB 49 7F :D3
43E0 BB 48 BD BA C2 86 03 B7 BB 49 C6 FF A6 C0 81 FE :2A
43F0 27 04 86 FF A7 5F 5A 26 F3 39 BD BB 4A BD BC F4 :91
---------------------------------------------------------
Sum: 73 7A ED 7C CB D0 35 2D 73 4D 32 95 6D 98 4D 31 :5D
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4400 8D 5D BD BC C3 86 FF BD BC CD BD BA 69 BE C5 B0 :04
4410 6F 84 C6 03 F7 BB 49 BD BA C2 E6 0E BD BA B1 33 :3F
4420 06 86 FF 1F 31 3A E6 84 A7 84 C1 C0 25 F5 CE 88 :9B
4430 00 FF BB 42 CC 01 03 B7 BB 44 F7 BB 49 7F BB 48 :FF
4440 BD BA B1, 30 01 C6 9D BD B9 3B C6 63 86 FF A7 C0 :82
4450 5A 26 FB 7F 88 00 BD BA C2 BD BA B1 6F 01 39 7F :0B
4460 C5 B2 7F C5 B7 CC 01 02 B7 BB 44 F7 BB 49 C6 03 :BB
4470 F7 BB 48 CE 88 00 FF BB 42 BD BA C2 FF C5 B0 31 :2A
4480 C4 A6 C4 26 04 8D 30 20 1F 43 27 2B 8E 02 DE A6 :FD
4490 80 A1 C0 26 13 8C 02 E6 26 F5 F7 C5 B2 A6 46 B7 :BA
44A0 C5 B3 A6 45 B7 C5 B4 39 33 A8 20 11 83 89 00 26 :0A
44B0 CB 5C C1 20 26 BA 39 B6 C5 B7 26 06 F7 C5 B7 FF :F1
44C0 C5 B5 39 7D C5 B2 26 FA C6 3F 7E BB 02 34 02 C6 :03
44D0 0F BD BA 9D 27 19 B6 BB 47 A1 01 26 12 B6 C5 B3 :23
44E0 A1 02 26 0B A6 84 A1 E4 27 05 C6 34 7E BB 02 5A :3E
44F0 2A DF 35 82 B6 02 E6 88 80 10 2B FE 03 B1 B8 0B :16
---------------------------------------------------------
Sum: 48 5C E9 BA BB F7 0D FF 3D 53 AD 2A 92 46 B1 86 :7B
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4500 10 22 FE 5F B7 BB 47 BD BA B1 26 B6 8E 88 00 BF :21
4510 BB 42 CC 00 02 B7 BB 44 F7 BB 49 F7 BB 48 BD BA :ED
4520 C2 B6 88 00 81 53 27 05 C6 47 7E BB 02 7C BB 44 :C3
4530 7F BB 48 BD BA C2 BD BA B1 33 01 8E 88 00 C6 9D :90
4540 7E B9 3B 9E D9 34 10 8D 5E 10 2B FD B3 B1 B8 0B :77
4550 10 22 FE 0F B7 BB 47 8D 39 B1 B8 0B 10 22 FE 03 :65
4560 B1 BB 47 10 26 FD 99 BD BC 5F 7D C5 B2 10 26 00 :81
4570 DC 35 10 9F D9 8D 30 BD BC 5F BD BC C3 8D 13 8E :98
4580 02 DE FE C5 B0 C6 08 BD B9 3B C6 03 F7 BB 49 7E :14
4590 BA C2 BD B9 FB 81 41 10 26 FD B5 BD B9 F5 81 53 :D6
45A0 10 26 FD AC BD B9 F5 10 8E CC 37 BD FC 20 7D 02 :43
45B0 DD 10 27 00 0B B6 02 E6 88 80 39 7D 02 DD 26 05 :85
45C0 C6 37 7E BB 02 7F C5 AF 96 BF 81 11 27 12 86 0F :E0
45D0 B7 C5 AF BD BA 97 27 08 4A 26 F5 C6 42 7E BB 02 :10
45E0 34 04 BD BC F4 BD BC 5F 35 04 C1 49 27 16 C1 4F :0D
45F0 27 4B C1 41 27 78 C1 52 27 05 C6 33 7E BB 02 C6 :4C
---------------------------------------------------------
Sum: A8 C1 B4 17 CD 01 AF 7F 68 D7 F3 CC C7 CA 9E F4 :51
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4600 4A 7E BB 02 86 10 B7 02 DA BD BC C3 BD BC CD BE :EE
4610 C5 B0 6D 0D 26 E4 EC 0B FD 02 DB BD BF 19 63 04 :C6
4620 BD C0 BE BD BA B1 6C 84 BD BA 97 B6 02 DA A7 84 :1E
4630 BA C5 AF 8A 80 BE 02 B4 D6 BF A7 85 39 86 20 B7 :03
4640 02 DA 7D C5 B2 27 1F 96 BF 81 11 27 05 C6 40 7E :AD
4650 BB 02 96 47 4C 26 F6 BD BC 05 B6 C5 B2 B7 C5 B7 :E0
4660 BE C5 B0 BF C5 B5 BD BF 40 BD BF 19 20 B5 86 20 :38
4670 B7 02 DA BD BC C3 86 FF BD BC CD 7D C5 B4 10 26 :C6
4680 FF 78 BD BF 19 F6 C5 B3 BD BA B1 30 06 34 10 3A :56
4690 1F 98 E6 84 AE E4 C1 C0 25 F5 32 62 BD BA 97 C4 :B4
46A0 3F ED 03 86 02 B7 BB 49 BD C0 5B 33 0F FF BB 42 :88
46B0 BD BA C2 CC 01 00 ED 0D BD C1 0A 10 83 01 00 26 :42
46C0 02 8D 03 7E BE 23 6F 0D 6F 0E A6 04 4C 81 08 26 :8F
46D0 01 4F 31 84 A7 04 E6 03 BD BA B1 3A 33 06 6C C4 :64
46E0 4D 26 06 8D 19 A7 C4 A7 23 7D BE FD 10 26 00 03 :C5
46F0 7E BC 2E BD BA B1 6D 01 26 02 6C 01 39 00 BD BA :43
---------------------------------------------------------
Sum: A0 CB 02 BF 67 38 1D D7 B3 AE F1 4E 70 B6 25 85 :2F
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4700 B1 30 06 CC 00 FF E1 80 27 0A 4C 81 98 25 F7 C6 :8B
4710 43 7E BB 02 C6 C0 E7 82 39 BD BA 97 10 8E 01 0F :62
4720 4F A7 80 31 3F 26 FA BD BA 97 B6 BB 47 A7 01 B6 :2A
4730 C5 B3 A7 02 A7 03 CC 01 00 ED 07 33 0F EF 09 39 :FF
4740 C6 41 B6 C5 B7 10 27 FB B9 B7 BB 48 C6 02 F7 BB :58
4750 49 BD BA C2 BE C5 B5 33 84 C6 20 4F A7 80 5A 26 :4D
4760 FB 8E 02 DE C6 08 BD B9 3B 7F BE FD FC 02 DB ED :E8
4770 43 27 06 10 83 02 00 26 09 D6 BF C1 11 26 03 F7 :BB
4780 BE FD BD BE FE B7 C5 B3 A7 46 C6 03 F7 BB 49 BD :D1
4790 BA C2 34 70 BD BA B1 6C 01 A6 01 B1 05 E5 25 03 :1F
47A0 BD BC 2E 35 F0 34 04 1F 89 C4 0F F7 C5 AF BD BA :61
47B0 97 E6 01 F7 BB 47 BD BA B1 6D 01 27 03 BD BC 2E :DE
47C0 BD BA 97 A6 84 81 20 26 2F EC 0D 10 83 01 00 27 :E2
47D0 17 33 8B 33 4F 86 1A 21 4F A7 C0 EC 0D C3 00 01 :8B
47E0 ED 0D 10 83 01 00 26 F0 BD C0 5B 33 0F FF BB 42 :BA
47F0 C6 03 F7 BB 49 BD BA C2 BD BA B1 6A 84 BE 02 B4 :87
---------------------------------------------------------
Sum: A8 19 A9 E7 ED 77 78 BE 75 47 CB C6 5F 80 D5 4F :3B
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4800 35 04 6F 85 BD BA 97 6F 84 39 FF B8 0E C4 0F BD :BC
4810 BA 9D 81 0D 26 02 6F 06 81 20 25 02 6C 06 34 02 :F2
4820 EC 0D C3 00 01 ED 0D 10 83 01 00 27 0F 30 8B 35 :71
4830 02 A7 0E 10 FE B8 0E 35 76 7E FC 1A 35 02 A7 89 :31
4840 01 0E E6 01 F7 BB 47 8D 12 33 0F FF BB 42 C6 03 :95
4850 F7 BB 49 BD BA C2 BD BE C6 20 D8 E6 03 54 54 CB :C9
4860 02 F7 BB 44 E6 03 C4 03 58 58 58 EB 04 F7 BB 48 :99
4870 39 FF B8 0E C4 0F BD BA 9D E6 0B 27 12 A6 0C 6F :30
4880 0B 7D B8 2C 26 EA 10 FE B8 0E 35 74 7E FC 1A EC :79
4890 0D 26 04 03 C0 20 EA 83 00 01 ED 0D 27 09 E6 05 :9D
48A0 6C 05 3A A6 0F 20 DA E6 05 4F C3 00 0F A6 8B 34 :CB
48B0 02 6F 05 A6 01 B7 BB 47 8D 04 35 02 20 C3 A6 04 :2B
48C0 4C 34 02 81 08 25 01 4F A7 04 E6 03 33 84 BD BA :42
48D0 B1 3A E6 06 30 C4 C1 C0 24 0A 35 02 80 08 26 12 :71
48E0 E7 03 20 DD C4 3F C1 08 10 24 FC 3C E0 E0 25 2B :2F
48F0 1F 98 34 02 86 02 B7 BB 49 BD C0 5B 33 0F FF BB :04
---------------------------------------------------------
Sum: 99 34 9A 93 B5 FB 6F 42 39 BA 5B 11 2C 18 8E DD :69
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4900 42 BD BA C2 8D 10 A6 E0 26 11 6D C9 00 FF 26 0C :3C
4910 33 5F 8D 10 26 F4 CC 01 00 ED 0D 39 A6 C9 00 FF :B7
4920 81 1A 26 F2 EC 0D 83 00 01 ED 0D 39 FF B8 0E C4 :EC
4930 0F BD BA 9D A6 06 5F 1F 01 86 0E 10 8E D0 E5 7E :B3
4940 BA 38 96 17 81 08 4A 4A 4A 2F 04 24 02 1A 02 39 :B4
4950 8E 04 3D C6 2C D7 11 8D E9 27 02 C6 20 8D 6F 81 :AB
4960 20 27 FA 81 22 26 0A C1 2C 26 06 1F 89 D7 11 20 :DD
4970 22 C1 22 27 11 81 0D 26 0D 8C 04 3D 27 44 A6 1F :FB
4980 81 0A 26 3E 86 0D 4D 27 17 91 11 27 1D 34 04 A1 :CC
4990 E0 27 17 A7 80 8C 05 3C 26 06 8D 3B 26 06 20 1E :70
49A0 8D 35 27 CD 6F 84 8E 04 3C 39 81 22 27 04 81 20 :1F
49B0 26 F2 8D 23 26 EE 81 20 27 F8 81 2C 27 E6 81 0D :E4
49C0 26 08 8D 13 26 DE 81 0A 27 DA 8D 18 20 D6 8D 07 :8D
49D0 27 09 C6 36 7E BB 02 8D 1A 0D C0 39 BE 02 B4 D6 :5E
49E0 BF E6 85 39 34 10 8D F4 C4 0F BD BA 9D A7 0C 63 :25
49F0 0B 35 90 34 74 0F C0 8D E3 10 2A F9 03 73 B8 2C :44
---------------------------------------------------------
Sum: BA 9B 6F 71 0C 60 F7 5D 1C 47 79 45 14 28 6C 9E :5C
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4A00 BD C0 74 7F B8 2C 35 F4 10 8E 99 BF BD FC 20 F1 :3D
4A10 B8 0B 10 22 F9 4D F7 BB 47 BD BD 07 BD BA B1 30 :0D
4A20 06 6F E2 C6 98 A6 80 43 26 02 6C E4 5A 26 F6 35 :41
4A30 04 10 8E 97 4C 7E BA 38 37 02 FF B8 0E 81 10 27 :AB
4A40 20 86 10 8D 40 5F 6D 0D 26 10 A6 0B AA 0E 26 0A :2B
4A50 CC FF FF 10 8E 97 4D 7E BA 38 10 8E 97 4C 7E BA :75
4A60 38 8D 22 A6 01 B7 BB 47 E6 02 8D 36 4A C4 3F 5C :9B
4A70 C1 3E 26 01 5F 34 04 6F E2 C6 08 3D E3 E1 10 8E :7B
4A80 97 4D 7E BA 38 34 02 BE 02 B4 E6 85 10 27 FA 38 :D2
4A90 10 2A F8 6C C4 0F BD BA 9D A6 84 A1 E0 10 26 FB :61
4AA0 59 39 BD BA B1 33 06 4F 4C 30 C4 3A E6 84 C1 C0 :A7
4AB0 25 F6 39 BD BC F4 BD BA 27 CC 01 02 B7 BB 44 F7 :DB
4AC0 BB 49 C6 03 F7 BB 48 8E 88 00 BF BB 42 BD BA C2 :D2
4AD0 34 10 10 8E D6 78 BD FC 20 1C BF 35 10 A6 84 27 :7A
4AE0 2B 43 27 38 34 10 8E C3 D3 BD BA 22 AE E4 6F 08 :D7
4AF0 BD B8 87 8E C3 DC BD B8 87 AE E4 BD C3 85 E6 0E :B0
---------------------------------------------------------
Sum: 60 94 3B 36 F0 07 B1 F1 70 3C 57 9F A0 9E 82 14 :74
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4B00 8D A0 8D 3C 8E C3 E3 BD B8 87 35 10 30 88 20 8C :CF
4B10 89 00 25 BC F6 BB 48 5C C1 20 25 A8 BD BA 27 BD :C8
4B20 BA B1 30 06 6F E2 C6 98 A6 80 43 26 02 6C E4 5A :8B
4B30 26 F6 35 02 8D 0A 8E C3 E8 BD B8 87 7F 05 AC 39 :88
4B40 34 04 C6 20 34 04 C6 64 8D 0B C6 0A 8D 07 8B 30 :37
4B50 BD B8 4A 35 86 34 04 5F A1 E4 25 05 A0 E4 5C 20 :C0
4B60 F7 32 61 CB 30 C1 30 26 04 E6 62 20 08 34 04 C6 :0E
4B70 30 E7 63 35 04 34 02 1F 98 BD B8 4A 35 82 8D 02 :A5
4B80 86 20 7E B8 4A 34 10 EC 0B 81 03 24 33 81 01 27 :E5
4B90 13 81 02 27 1F 6D 0D 26 27 8E C3 FA 5D 26 1C 8E :1B
4BA0 C4 03 20 17 5D 27 19 E6 0D 8E C4 0C 5D 26 0C 8E :09
4BB0 C4 15 20 07 EA 0D 26 08 8E C4 1E BD B8 87 35 90 :56
4BC0 BD C3 40 8D BB 86 42 AB 0C 8D B3 86 53 AB 0D 8D :E5
4BD0 AD 35 90 20 20 20 20 20 20 20 22 00 22 20 3A 20 :10
4BE0 27 20 00 20 27 0D 0A 00 20 43 6C 75 73 74 65 72 :A7
4BF0 73 20 66 72 65 65 2E 0D 0A 00 42 41 53 20 41 53 :04
---------------------------------------------------------
Sum: 33 0D E1 91 85 84 71 54 F4 C7 85 01 B8 07 9A 39 :53
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4C00 43 20 00 42 41 53 20 42 49 4E 20 00 44 41 54 20 :4B
4C10 52 44 4D 20 00 44 41 54 20 53 45 51 20 00 4F 42 :96
4C20 4A 20 20 20 20 20 00 BE 02 B4 A6 88 11 2B 0B 86 :59
4C30 11 97 BF 10 8E CE 4B BD FC 20 0F BF 0F 15 7F 02 :6A
4C40 D7 34 14 8E 05 A0 C6 01 E7 84 6F 02 9D DE 35 14 :B9
4C50 10 8E D6 B5 BD FC 20 9E 47 9F AD 30 01 27 04 30 :BF
4C60 1F 9F A6 9E E9 27 04 96 65 A7 84 B6 02 F3 27 07 :15
4C70 10 8E 8F 39 BD FC 20 8E FC 04 10 DE AF 10 8E 8F :97
4C80 91 7E B9 99 FF B8 0E 9E 4F 9F B2 9E AD 30 01 27 :07
4C90 19 30 1F 9F 49 9E 4F 9F 4D 9E B4 27 0D 96 B1 26 :1C
4CA0 09 03 B1 10 8E C6 5D 7E BA 38 10 8E 9B 47 BD FC :27
4CB0 20 D6 AC C1 41 25 04 C1 4A 23 07 10 8E 8E 35 7E :E1
4CC0 BA 38 C0 40 8E C4 ED 5A 27 06 A6 80 26 FC 20 F7 :17
4CD0 34 30 10 8E EC 05 BD FC 20 86 07 BD B8 4A BD BA :8F
4CE0 27 35 30 BD BA 22 10 8E 8E 69 7E BA 38 44 69 72 :49
4CF0 65 63 74 6F 72 79 20 46 75 6C 6C 00 54 6F 6F 20 :9B
---------------------------------------------------------
Sum: 53 91 F4 AF 14 E9 4E 7A E0 3C DE B8 20 1D 74 CE :7D
```

リスト1　SYSGENプログラム　ダンプ・リスト

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4D00 4D 61 6E 79 20 4F 70 65 6E 20 44 69 73 6B 20 46 :58
4D10 69 6C 65 73 00 44 69 73 6B 20 46 75 6C 6C 00 46 :31
4D20 69 65 6C 64 20 4F 76 65 72 66 6C 6F 77 00 53 74 :D9
4D30 72 69 6E 67 20 4E 6F 74 20 46 69 65 6C 64 65 64 :CE
4D40 00 42 61 64 20 52 65 63 6F 72 64 20 4E 75 6D 62 :38
4D50 65 72 00 42 61 64 20 46 69 6C 65 20 53 74 72 75 :4C
4D60 63 74 75 72 65 00 44 72 69 76 65 20 4E 6F 74 20 :8E
4D70 52 65 61 64 79 00 44 69 73 6B 20 57 72 69 74 65 :AB
4D80 20 50 72 6F 74 65 63 74 65 64 00 44 49 53 4B 20 :15
4D90 42 41 53 49 43 20 46 65 61 74 75 72 65 20 28 52 :E8
4DA0 61 6E 64 6F 6D 20 46 69 6C 65 29 00 00 00 00 00 :D8
4DB0 00 00 00 E5 E5 E5 E5 E5 C6 3F 20 02 C6 5D 34 06 :FD
4DC0 1A 50 7F FD 0F F7 B8 10 32 61 E6 E0 C0 3F 54 BD :1D
4DD0 B8 D4 7D FD 0F 1C AF 39 10 EF 8C 19 10 FE B8 0E :91
4DE0 7D FD 0F 1C AF AD A4 1A 50 34 01 7F FD 0F 35 01 :05
4DF0 10 EE 8C 01 39 EE 8C 8D C7 8D C5 8D C3 8D C1 8D :0F
-------------------------------------------------------------
Sum: CD 36 A4 56 CE 1E 36 4C 70 38 A3 26 27 A5 48 91 :81
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4E00 BF 8D BD 8D BB 8D B9 8D B7 8D B5 8D B3 8D B1 8D :28
4E10 AF 8D AD 8D AF 8D AD E5 E5 E5 E5 E5 E5 E5 E5 E5 :6C
4E20 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4E30 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4E40 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4E50 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4E60 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4E70 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4E80 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4E90 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4EA0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4EB0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4EC0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4ED0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4EE0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
4EF0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
-------------------------------------------------------------
Sum: F4 A0 F0 A0 F0 A0 EC F8 22 F8 20 F8 1E F8 1C F8 :F4
```

リスト2　スーパーモニタ，ローダープログラム　ダンプ・リスト

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5000 1A 50 7F FD 0F 8E 50 20 CE D0 00 10 8E 1C 00 A6 :F1
5010 84 6F 80 A7 C0 31 3F 26 F6 7D FD 0F 39 00 00 00 :28
5020 7E D3 C1 7E D3 FD D5 52 E9 FB D5 FF D7 94 D9 F4 :77
5030 D6 B5 D6 43 D0 3A D5 52 D5 52 D5 52 E3 92 D7 49 :B8
5040 D5 52 D5 52 D0 A1 D0 8A E2 2B E1 00 D5 52 D5 52 :55
5050 D5 52 E2 52 D8 91 D5 52 D5 52 96 C3 8E D0 CA 81 :14
5060 28 27 20 A6 84 27 34 BD B8 23 C6 29 8D 28 34 10 :74
5070 8E D0 85 E7 01 BD B8 23 35 10 BD B8 23 8D 17 8D :71
5080 1B 20 E0 A6 84 27 14 81 45 27 05 BD B8 23 8D 0C :A3
5090 8D 04 8D 02 20 ED 5A A6 80 26 FB 39 86 0D 8D 02 :29
50A0 86 0A 7E B8 1D 13 00 20 00 00 7F 05 AC 7F 05 A9 :73
50B0 CE CF 06 6F 43 DC AF ED 4A CC 8F E1 ED 4C 7E D6 :E0
50C0 56 8E D0 AF 73 CF 14 26 03 8E D0 BC 7E B8 23 50 :A5
50D0 72 69 6E 74 65 72 20 4F 4E 0D 0A 00 50 72 69 6E :01
50E0 74 65 72 20 4F 46 46 0D 0A 00 42 20 7B 3C 61 64 :3B
50F0 64 72 65 73 73 3E 2C 5B 53 2C 52 5D 7D 00 62 72 :65
-------------------------------------------------------------
Sum: EE AD F8 1B 3D D4 8D B7 E3 2A 1D 29 31 7A 86 74 :FB
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5100 65 61 6B 20 70 6F 69 6E 74 20 64 69 73 70 6C 61 :18
5110 79 2C 73 65 74 20 6F 72 20 72 65 73 65 74 2E 00 :63
5120 43 20 3C 61 64 64 72 65 73 73 3E 00 63 61 6C 6C :5F
5130 20 73 75 62 72 6F 75 74 69 6E 65 2E 00 44 20 7B :7D
5140 7B 3C 73 74 61 72 74 3E 7D 7B 2C 3C 65 6E 64 3E :F8
5150 7D 7D 00 64 75 6D 70 20 6D 65 6D 6F 72 79 2E 00 :97
5160 45 20 7B 3C 61 64 64 72 65 73 73 3E 7D 00 65 64 :86
5170 69 74 20 6D 65 6D 6F 72 79 2E 00 46 20 3C 73 74 :4D
5180 61 72 74 3E 2C 3C 65 6E 64 3E 2C 3C 63 6F 6E 73 :7D
5190 74 2E 3E 00 66 69 6C 6C 20 6D 65 6D 6F 72 79 2E :6E
51A0 00 47 20 7B 3C 61 64 64 72 65 73 73 3E 7D 00 67 :26
51B0 6F 20 75 73 65 72 20 70 72 6F 67 72 61 6D 2E 00 :94
51C0 48 00 68 65 6C 70 2E 00 4C 20 7B 7B 3C 73 74 61 :05
51D0 72 74 3E 7D 7B 2C 3C 65 6E 64 3E 7D 7D 00 64 69 :C0
51E0 73 61 73 73 65 6D 62 6C 65 2E 00 4D 20 3C 73 74 :7D
51F0 61 72 74 3E 2C 3C 65 6E 64 3E 2C 3C 74 6F 70 3E :5B
-------------------------------------------------------------
Sum: B9 BB 71 88 01 CF FC E8 23 63 C8 48 6D 95 60 E2 :FB
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5200 00 6D 6F 76 65 20 6D 65 6D 6F 72 79 20 62 6C 6F :CD
5210 63 6B 2E 00 50 00 70 72 69 6E 74 65 72 20 6F 6E :4D
5220 20 6F 66 66 20 74 6F 67 67 6C 65 2E 00 51 00 72 :EE
5230 65 74 75 72 6E 20 74 6F 20 62 61 73 69 63 2E 00 :81
5240 52 20 3C 64 72 76 3E 2C 3C 74 72 6B 73 65 63 3E :6A
5250 2C 3C 73 74 61 72 74 3E 2C 3C 65 6E 64 3E 00 72 :23
5260 65 61 64 20 66 72 6F 6D 20 64 69 73 6B 2E 00 53 :4A
5270 20 3C 73 74 61 72 74 3E 2C 3C 65 6E 64 3E 2C 3C :0D
5280 63 6F 6E 73 74 2E 3E 7B 2C 3C 63 6F 6E 73 74 2E :CB
5290 3E 2C 2E 2E 7D 00 73 65 61 72 63 68 20 63 6F 6E :19
52A0 73 74 61 6E 74 2E 00 57 20 3C 64 72 76 3E 2C 3C :FD
52B0 74 72 6B 73 65 63 3E 2C 3C 73 74 61 72 74 3E 2C :CA
52C0 3C 65 6E 64 3E 00 77 72 69 74 65 20 74 6F 20 64 :63
52D0 69 73 6B 2E 00 58 20 7B 5B 43 43 2C 41 2C 42 2C :50
52E0 44 50 2C 49 58 2C 49 59 2C 55 53 2C 53 50 2C 50 :4E
52F0 43 5D 7D 00 72 65 67 69 73 74 65 72 20 64 69 73 :E2
-------------------------------------------------------------
Sum: 9F BA E8 17 AF 28 8B D4 5D D8 4F CD 3F 1C DC E5 :FB
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5300 70 6C 61 79 20 6F 72 20 63 68 61 6E 67 65 2E 00 :6B
5310 3C 63 6F 6E 73 74 2E 3E 3A 28 68 65 78 2D 64 65 :6C
5320 63 69 6D 61 6C 29 6F 72 28 22 2E 22 2B 63 68 61 :01
5330 72 61 63 74 65 72 29 00 20 00 3C 74 72 6B 73 65 :2F
5340 63 3E 3A 28 3C 74 72 6B 3E 2C 3C 73 65 63 3E 29 :D8
5350 6F 72 28 22 23 22 2B 3C 63 6C 75 73 74 65 72 3E :17
5360 29 00 20 00 00 00 2D 2D 20 55 52 41 44 4F 53 20 :B1
5370 73 75 70 65 72 20 6D 6F 6E 69 74 6F 72 20 49 2F :EF
5380 4F 20 76 65 72 73 69 6F 6E 20 2D 2D 20 20 2D 2D :89
5390 20 43 6F 70 79 72 69 67 68 74 20 28 43 29 31 39 :F7
53A0 38 33 20 62 79 20 44 2E 4F 2E 53 2E 20 20 2D 2D :90
53B0 0D 0A 00 20 42 79 20 54 6F 6D 2E 57 61 74 61 6E :6B
53C0 61 62 65 20 00 0D 0A 49 6C 6C 65 67 61 6C 20 77 :B0
53D0 72 69 74 65 20 61 64 64 72 65 73 73 2E 07 0D 0A :06
53E0 00 33 42 37 20 FF B8 0E CE FC 70 FF FF FA CE FC :8D
53F0 60 8E D4 48 C6 20 8D 4F CE CF 06 1A 80 1F A8 A7 :77
-------------------------------------------------------------
Sum: D6 EA 86 C6 E1 3F 58 75 22 D3 C6 CC FD 00 48 06 :CB
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5400 C4 6F 43 FC B8 0E ED 4A 10 AF 4C 8E D3 46 BD B8 :96
5410 23 7F CF 14 CE FF F0 BD EA 82 7E D5 1D 30 E4 10 :FF
5420 CE B7 F0 CE CF 06 C6 0A 8D 1D EC 81 83 00 01 ED :70
5430 42 AF C4 BF B8 0E 1F 03 BD EA 82 8E D4 2F BD B8 :8B
5440 23 BD D8 9D 7E D5 1D A6 80 A7 C0 5A 26 F9 39 53 :57
5450 74 6F 70 20 61 74 20 62 72 65 61 6B 20 70 6F 69 :D5
5460 6E 74 2E 2E 2E 0D 0A 00 7D FD 0F A6 C0 7F FD 0F :FD
5470 39 E5 E5 E5 E5 E5 E5 E5 7F FD 0F 7E D0 03 7D FD :D2
5480 0F 3B 3F 34 60 73 05 A9 26 08 73 05 A9 8E D5 0D :FD
5490 20 1A 86 05 BD B8 1A 10 8E D9 3D BD B8 29 10 8E :44
54A0 DA F6 BD B8 29 8E D5 15 6F 04 6F 05 7F 03 13 9D :FF
54B0 DE FC 03 12 10 26 FB F2 B6 04 3D 27 0C 81 01 10 :CE
54C0 27 FB E7 7F 05 A9 7E D5 21 EC 04 83 00 04 22 0E :51
54D0 26 07 B6 05 A9 27 07 20 B4 4F 5F B7 05 A9 C3 04 :6D
54E0 40 34 06 8E 04 41 CE 04 3D AC E4 22 12 A6 80 81 :C7
54F0 20 24 02 86 20 11 83 05 3C 27 EE A7 C0 20 EA 32 :79
-------------------------------------------------------------
Sum: C9 7A 4B 08 27 5D B3 BF 59 35 08 4C E0 3E C9 42 :97
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5500 62 11 83 04 3C 23 06 A6 C2 81 20 27 F4 B6 05 A9 :E7
5510 27 0A B6 04 3D 81 24 27 0F 7F 05 A9 6F 41 86 0D :73
5520 BD B8 1A 86 0A BD B8 1A 8E 04 3D 35 E0 13 00 04 :A9
5530 3D 00 00 01 04 12 00 04 3D 00 00 01 04 10 FF CF :78
5540 17 10 FE CF 17 7F 05 AC 7D 05 A9 26 06 8E D5 5A :4F
5550 BD B8 23 BD D4 63 A6 80 27 E7 81 24 10 27 01 EA :87
5560 84 DF 80 41 25 0C 81 19 22 08 CE D0 06 48 AD D6 :88
5570 20 CF 8E D5 5E BD B8 23 20 C7 0D 0A 3E 00 3F 3F :02
5580 3F 20 63 6F 6D 6D 61 6E 64 20 65 72 72 6F 72 2E :B6
5590 07 0D 0A 00 3F 3F 20 70 61 72 61 6D 65 74 65 72 :7D
55A0 20 65 72 72 6F 72 2E 07 0D 0A 00 8E D5 74 BD B8 :E2
55B0 23 20 8E 6F E2 6F E2 8D 25 C1 2C 27 1E 5D 27 1B :F6
55C0 BD D5 C5 25 E6 68 61 69 E4 68 61 69 E4 68 61 69 :C0
55D0 E4 68 61 69 E4 EA 61 E7 61 20 DC 5D 35 86 E6 80 :07
55E0 C1 20 27 FA 39 C0 30 25 16 C1 0A 25 10 CB 30 C4 :25
55F0 DF C0 41 25 0A C0 06 C0 FA 25 04 CB 0A 1C FE 39 :E0
-------------------------------------------------------------
Sum: C5 18 7D 2E FF 7D 4F FA CE 8A A4 74 9E A0 7C 3B :B2
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5600 8D DC 30 1F C1 2C 27 01 5D 39 8D F4 27 9D 7E D5 :FB
5610 93 8D F7 10 27 FF 94 39 8D F0 10 26 FF 8D 39 8D :1F
5620 F7 FD CF 19 BD D6 6B CC D6 28 FD D0 04 8E CF 06 :D8
5630 EE 0A 33 52 32 C4 C6 0A BD D4 27 FC CF 19 ED C1 :8D
5640 CC FC 7A ED C4 7E FC 76 30 E4 10 CE B7 F0 CE CF :19
5650 06 C6 0A BD D4 27 30 02 AF C4 CC D3 FD FD D0 04 :A0
5660 7E D5 1D BD D5 E0 5D 27 08 BD D5 F8 CE CF 06 ED :88
5670 4C 8D 18 BD EA C7 8E CF 06 EE 0A 33 54 32 C4 C6 :FD
5680 0A BD D4 27 EC 02 ED C4 7E FC 76 8E D6 82 BD B8 :AC
5690 23 BD B8 17 84 DF 81 59 10 26 FE A5 BD B8 1A 7E :D2
56A0 D0 7C 41 72 65 20 79 6F 75 20 73 75 72 65 20 28 :08
56B0 59 2F 61 6E 79 29 20 3F 20 00 BD D5 BE C1 2E 26 :DD
56C0 09 E6 80 A6 84 10 26 FE E2 39 30 1F BD D5 F8 4D :0E
56D0 10 26 FE D7 39 BD D5 F1 FD CF 1B BD D5 F1 10 B3 :F4
56E0 CF 1B 10 25 FE C5 FD CF 1D 8D CF BE CF 1B 1F 98 :86
56F0 8D 13 BC CF 1D 23 F9 39 34 06 CC A6 C0 FD FC 63 :65
-------------------------------------------------------------
Sum: 6C F3 5A 4D 54 F0 FB 40 BD 55 06 6F B3 FD 23 2E :0D
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5700 35 06 7E FC 60 34 06 CC A7 80 FD FC 63 35 06 8C :65
5710 80 00 10 25 25 6A 7D CF 42 27 09 BD E0 A3 CC 01 :0F
5720 13 BD DB 93 8E D3 A5 BD B8 23 7E D5 21 8D 02 1F :FE
5730 98 E6 80 8D 0D 58 58 58 58 34 04 E6 80 8D 03 EA :10
5740 E0 39 BD D5 C5 10 25 FD F8 39 8D E1 1F 03 A6 80 :89
5750 81 3A 10 26 FD EB 8D D9 1E 13 1F 98 8D A7 1E 13 :8C
5760 A6 80 81 20 27 F0 7E D5 21 BD D5 F1 FD CF 1F BD :7D
5770 D5 F1 10 B3 CF 1F 10 25 FE 31 FD CF 21 BD D5 F8 :52
5780 1F 01 FE CF 1F 10 B3 CF 1F 22 0D BD D6 D8 BD D6 :EA
5790 E5 11 B3 CF 21 23 F4 39 F3 CF 21 B3 CF 1F 1F 01 :8D
57A0 FE CF 21 BD D6 D8 BD D6 E5 30 1E 33 5E 11 B3 CF :43
57B0 1F 24 F0 39 CC FF FF FD CF 25 B6 CF 14 B7 05 AC :28
57C0 BD D5 E0 27 12 BD D5 93 34 06 27 06 BD D5 F8 FD :BE
```

リスト2　スーパーモニタ，ローダープログラム　ダンプ・リスト

```
57D0 CF 25 35 06 FD CF 23 FE CF 23 11 B3 CF 25 22 D3 :BB
57E0 8E CB 00 86 24 A7 80 1F 30 8D 5C 86 3A A7 80 D6 :1F
57F0 C3 C4 60 54 54 34 44 BD D6 D8 8D 54 86 20 A7 80 :20
------------------------------------------------------------
Sum: 3A 1B 7E AA 41 44 DF C8 FD 0C 29 B2 11 A8 64 56 :00


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5800 11 B3 CF 25 22 03 5A 26 EE 35 24 C1 08 27 22 33 :E9
5810 A4 86 3B A7 80 BD D6 D8 81 7F 27 04 81 20 24 02 :E9
5820 86 2E A7 80 11 B3 CF 25 22 03 5A 26 E8 86 3B A7 :88
5830 80 CC 0D 0A ED 81 6F 84 FF CF 23 8E CB 00 BD B8 :83
5840 23 BD D8 56 27 91 39 8D 07 1E 89 8D 03 1E 89 39 :AA
5850 34 02 44 44 44 44 8D 06 A6 E4 8D 02 35 82 84 0F :3C
5860 81 0A 25 02 8B 07 8B 30 A7 80 6F 84 39 34 10 8E :24
5870 D8 70 9D DE 35 90 8D F5 B6 CF 16 27 12 B6 CF 15 :78
5880 81 1B 27 0B 81 20 10 26 00 05 BD B8 17 81 20 39 :10
5890 15 00 CF 15 00 00 34 06 34 02 C6 08 4F 68 E4 89 :5B
58A0 30 A7 80 5A 26 F6 35 02 A7 84 35 86 86 20 A7 80 :B7
58B0 39 BD D5 E0 10 26 00 8A 5D 10 26 FC EE B6 CF 14 :81
58C0 B7 05 AC 8E D8 ED BD B8 23 8E CB 00 CE CF 06 A6 :F5
58D0 C4 BD D8 30 8D D6 A6 C0 8D BC 8D D0 C6 03 A6 C0 :27
58E0 BD D8 30 8D C7 5A 26 F6 EC 5D BD D8 27 8D BD C6 :A4
58F0 05 34 04 EC C1 BD D8 27 8D B2 6A E4 26 F5 32 61 :E1
------------------------------------------------------------
Sum: A7 B9 9F 61 6F 76 26 AC FB CB C0 81 7A 6A 3F 62 :A3


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5900 CC 0D 0A ED 81 6F 84 8E CB 00 7E B8 23 43 43 20 :9C
5910 2E 46 48 49 4E 5A 56 43 20 41 2E 20 42 2E 20 44 :C9
5920 50 20 44 2E 2E 2E 20 49 58 2E 2E 20 49 59 2E 2E :79
5930 20 55 53 2E 2E 20 53 50 2E 2E 20 50 43 2E 2E 0D :5F
5940 0A 00 10 8E D9 C6 33 84 34 20 A6 C0 84 DF A1 A4 :60
5950 26 0C 31 21 4D 26 F3 35 20 CE CF 06 20 10 A6 A0 :58
5960 26 FC 32 62 31 21 A6 A4 43 26 DB 7E D5 8B 8E CB :CD
5970 00 A6 A0 27 04 A7 80 20 F8 86 3A A7 80 E6 A0 C1 :DE
5980 04 24 31 A6 C5 BD D8 30 CC 3D 00 ED 81 8E CB 00 :59
5990 34 20 BD B8 23 35 20 BD D4 63 A6 84 27 D0 81 2E :05
59A0 27 41 BD D5 F8 4D 10 26 FC 01 A6 3F 26 02 CA 80 :C9
59B0 E7 C6 20 BA 34 20 31 C5 EC A4 BD D8 27 CC 3D 00 :26
59C0 ED 81 8E CB 00 34 20 BD B8 23 35 20 BD D4 63 A6 :A2
59D0 84 27 09 81 2E 27 0C BD D5 F8 ED A4 35 20 A6 A4 :50
59E0 43 26 8B 7E D5 21 43 43 00 00 41 00 01 42 00 02 :74
59F0 44 50 00 03 49 58 00 04 49 59 00 06 55 53 00 08 :94
------------------------------------------------------------
Sum: FE DF E9 84 E6 FE 41 80 5E F0 F0 85 27 0D 90 71 :E7


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5A00 53 50 00 0A 50 43 00 0C FF 8E CF 51 C6 20 6F 80 :CE
5A10 5A 26 FB 39 96 C3 81 50 10 26 FB 56 BD D5 BE 5D :12
5A20 27 08 30 1F BD D5 F8 FD CF 27 86 0C BD B8 1A 7F :9B
5A30 CF 2A 7F CF 29 7F CF 2B 7F 03 13 BD E0 A8 8E DF :30
5A40 F7 BD B8 23 8D C3 FE CF 27 5F 34 04 1F 30 8E CB :12
5A50 00 BD D8 27 CC 3A 20 ED 81 C6 10 6F E2 BD D6 D8 :E2
5A60 BD D8 30 AB E4 A7 E4 86 20 A7 80 5A 26 EF 33 50 :9E
5A70 86 3A A7 80 A6 E4 BD D8 30 CC 20 3B ED 81 10 8E :69
5A80 CF 51 5F BD D6 D8 34 02 AB A5 A7 A5 35 02 81 7F :F3
5A90 27 04 81 20 24 02 86 2E A7 80 5C C1 10 26 E4 86 :8A
5AA0 3B A7 80 CC 0D 0A ED 81 6F 84 8E CB 00 BD B8 23 :97
5AB0 35 06 10 8E CF 61 A7 A5 5C C1 10 26 8D 8E E0 52 :F5
5AC0 CE CF 51 6F E2 C6 10 A6 C0 BD D8 30 AB E4 A7 E4 :5A
5AD0 86 20 A7 80 5A 26 F0 86 3A A7 80 A6 E0 B7 CF 2C :5C
5AE0 BD D8 30 CC 0D 0A ED 81 6F 84 8E E0 47 BD B8 23 :56
5AF0 10 8E DA F6 BD B8 29 BD DC 78 BD B8 17 81 7F 10 :B9
------------------------------------------------------------
Sum: 64 8B 83 8E 8B D5 6B 5E B7 40 8B 3D EF FE 26 79 :74


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5B00 27 01 7C 81 20 25 53 F6 CF 2B 26 07 81 2E 26 08 :B7
5B10 BD B8 17 7F CF 2A 20 23 BD E0 E2 1F 89 BD D5 C5 :C5
5B20 25 38 86 F0 73 CF 2A 27 06 58 58 58 58 86 0F 34 :95
5B30 06 BD DC 52 BD D6 D8 A4 E0 AA E0 BD DC 52 30 C4 :49
5B40 34 02 BD D6 D8 1F 89 35 02 BD D6 E5 BD DC B5 7D :C3
5B50 CF 2A 26 A3 73 CF 2A 7E DB D5 CE DB 4D A1 C0 27 :DA
5B60 08 33 42 6D C4 26 F6 20 8E EE C4 6E C4 48 DD 65 :E6
5B70 68 DD 65 03 DB 84 1B DB 84 19 DB 98 1A DB A4 06 :B1
5B80 DD 51 02 DD 5D 08 DB EC 12 DC 6A 0B DB B6 0D DB :15
5B90 C7 09 DC 41 1C DB D5 1D DB EC 1E DC 02 1F DC 0F :A3
5BA0 11 DB B0 00 7F 03 13 BD E0 A3 CC 01 13 BD DB 93 :7C
5BB0 7E D5 21 34 30 7E DC A2 FC CF 27 83 01 00 FD CF :16
5BC0 27 7E DA 1E FC CF 27 C3 01 00 FD CF 27 7E DA 1E :BC
5BD0 73 CF 2B 7E DA D7 7F CF 2A 86 F0 7D CF 29 27 01 :27
5BE0 4F B7 CF 29 7E DA D7 7F CF 2A B6 CF 29 84 F0 B7 :7E
5BF0 CF 29 7E DC 0F 7D CF 2B 26 05 73 CF 2A 26 0A B6 :55
------------------------------------------------------------
Sum: 6D 21 80 1E 94 ED 24 36 4A 95 14 56 60 46 EC AC :8E


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5C00 CF 29 8B 01 B7 CF 29 25 30 7E DA D7 7D CF 2B 26 :54
5C10 05 73 CF 2A 27 F3 B6 CF 29 80 01 B7 CF 29 25 2D :BB
5C20 20 E7 B6 CF 29 80 10 B7 CF 29 25 21 7E DA D7 B6 :1F
5C30 CF 29 8B 10 B7 CF 29 24 F3 B6 CF 29 80 10 B7 CF :1D
5C40 29 FC CF 27 C3 00 10 FD CF 27 7E DA 1E B6 CF 29 :05
5C50. 8B 10 B7 CF 29 FC CF 27 83 00 10 FD CF 27 7E DA :1A
5C60 1E 7F CF 2A B6 CF 29 84 F8 8B 08 B7 CF 29 25 C9 :F0
5C70 20 BA 34 14 BE CF 27 F6 CF 29 3A 33 84 35 94 FC :7A
5C80 CF 27 4C 5F FD CF 27 7E DA 1E FC CF 27 C3 00 FF :BE
5C90 5F 4A FD CF 27 7E DA 1E 34 30 B6 CF 29 84 0F 7D :34
5CA0 CF 2B 27 04 8B 3B 20 0E C6 03 8B 02 3D 34 04 B6 :9A
5CB0 CF 2A 84 01 AB E0 34 02 F6 CF 29 54 54 54 54 5C :D9
5CC0 35 02 8E DC AE ED 01 6F 03 BD B8 20 35 B0 12 00 :3B
5CD0 00 00 00 00 00 34 06 8E DC B1 BD D8 30 8E DC AE :32
5CE0 B6 CF 29 84 0F C6 03 8B 02 3D 1F 98 F6 CF 29 54 :CD
5CF0 54 54 54 5C 34 06 ED 01 BD B8 20 86 12 A7 02 F6 :4C
------------------------------------------------------------
Sum: C0 DC 23 2D 69 00 93 A2 9C 3B B9 A3 D8 A0 64 26 :BF


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5D00 CF 29 C4 0F CE CF 51 33 C5 A6 62 AB C4 A0 63 A7 :D2
5D10 C4 30 03 BD D8 30 8E DC AE BD B8 20 86 37 E6 61 :6D
5D20 ED 01 CE CF 61 5A 33 C5 A6 62 AB C4 A0 63 A7 C4 :23
5D30 30 03 BD D8 30 8E DC AE BD B8 20 86 12 A7 02 B6 :9C
5D40 CF 2C AB 62 A0 63 B7 CF 2C 30 03 BD D8 30 8E DC :1F
5D50 AE BD B8 20 35 06 B6 CF 29 84 0F 8B 3B BD DB 93 :B0
5D60 A6 E4 81 7F 27 04 81 20 24 02 86 2E BD B8 1A 35 :F4
5D70 86 7F CF 2A 7E DB DF 7F CF 2A 7E DA D7 7D CF 2A :53
5D80 26 F5 7E DB F6 8E DD 71 BD B8 23 BD B8 17 7E DA :C2
5D90 1E 0C 3C 74 65 6E 20 6B 65 79 3E 20 20 3C 63 6F :A2
5DA0 6E 74 72 6F 6C 20 6B 65 79 3E 0D 0A 20 41 20 42 :B0
5DB0 20 43 20 44 20 20 48 4F 4D 45 20 28 5E 4B 29 3A :84
5DC0 20 63 75 72 73 6F 72 20 68 6F 6D 65 20 6F 72 20 :A8
5DD0 62 6F 74 74 6F 6D 0D 0A 20 37 20 38 20 39 20 45 :19
5DE0 20 20 44 55 50 20 20 28 5E 51 29 3A 20 68 65 78 :08
5DF0 20 6D 6F 64 65 20 3C 2D 2D 3E 20 63 68 72 20 6D :A3
------------------------------------------------------------
Sum: ED C0 ED 3F 2F 87 46 CE 19 46 5F AE C1 64 85 5F :18


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5E00 6F 64 65 0D 0A 20 34 20 35 20 36 20 46 20 20 54 :48
5E10 41 42 20 20 28 5E 49 29 3A 20 6D 6F 76 65 20 38 :24
5E20 20 62 79 74 65 73 20 74 61 62 0D 0A 20 31 20 32 :58
5E30 20 33 20 20 20 20 49 4E 53 20 20 28 5E 52 29 3A :38
5E40 20 73 63 72 6F 6C 6C 20 62 61 63 6B 20 70 61 67 :B8
5E50 65 0D 0A 20 20 30 20 20 2E 20 20 20 20 44 45 4C :AF
5E60 20 20 20 20 20 20 3A 20 73 63 72 6F 6C 6C 20 66 :2F
5E70 6F 72 77 61 72 64 20 70 61 67 65 0D 0A 20 20 20 :C3
5E80 20 20 20 20 20 20 20 43 52 20 20 20 28 5E 4D 29 :D1
5E90 3A 20 6D 6F 76 65 20 74 6F 70 20 6F 66 20 6E 65 :6C
5EA0 78 74 20 6C 69 6E 65 0D 0A 20 3C 63 75 72 73 6F :53
5EB0 72 20 6B 65 79 20 61 6E 64 20 6F 74 68 65 72 3E :AE
5EC0 0D 0A 53 48 49 46 54 20 2B 20 75 70 20 20 20 20 :65
5ED0 20 28 5E 59 29 3A 20 6D 6F 76 65 20 62 61 63 6B :EA
5EE0 20 24 31 30 30 20 62 79 74 65 73 0D 0A 53 48 49 :17
5EF0 46 54 20 2B 20 64 6F 77 6E 20 20 20 28 5E 5A 29 :26
------------------------------------------------------------
Sum: DB CB 3C 30 12 48 17 8A 32 F8 82 EB 0F CF 34 69 :1F


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5F00 3A 20 6D 6F 76 65 20 66 6F 72 77 61 72 64 20 24 :6A
5F10 31 30 30 20 62 79 74 65 73 0D 0A 53 48 49 46 54 :6D
5F20 20 2B 20 6C 65 66 74 20 20 20 28 5E 42 29 3A 20 :C1
5F30 6D 6F 76 65 20 62 61 63 6B 20 62 79 74 65 0D 0A :53
5F40 53 48 49 46 54 20 2B 20 72 69 67 68 74 20 20 28 :6F
5F50 5E 46 29 3A 20 6D 6F 76 65 20 66 6F 72 77 61 72 :8F
5F60 64 20 62 79 74 65 0D 0A 42 53 20 20 20 20 20 20 :A4
5F70 20 20 20 20 20 20 20 28 5E 48 29 3A 20 6D 6F 76 :83
5F80 65 20 63 75 72 73 6F 72 20 6C 65 66 74 0D 0A 22 :27
5F90 2E 22 2B 6B 65 79 20 28 65 78 2E 43 52 2C 65 74 :B1
5FA0 63 29 3A 20 73 65 74 20 41 53 43 49 49 20 63 6F :AD
5FB0 64 65 20 6F 66 20 6B 65 79 28 68 65 78 20 6D 6F :90
5FC0 64 65 29 0D 0A 45 53 43 20 20 20 20 28 5E 5B 29 :6E
5FD0 20 6F 72 20 28 5E 43 29 3A 20 65 78 69 74 0D 0A :3E
5FE0 48 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 :28
5FF0 20 20 20 3A 20 68 65 6C 70 28 68 65 78 20 6D 6F :CC
------------------------------------------------------------
Sum: 73 9C EA 6F 87 54 B9 2D 0D CA 6C 30 46 EA F1 08 :C5


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6000 64 65 29 0D 0A 0D 0A 20 70 72 65 73 73 20 61 6E :5C
6010 79 20 6B 65 79 2E 00 0B 61 64 72 73 3A 20 2B 30 :7A
6020 20 2B 31 20 2B 32 20 2B 33 20 2B 34 20 2B 35 20 :96
6030 2B 36 20 2B 37 20 2B 38 20 2B 39 20 2B 41 20 2B :C1
6040 42 20 2B 43 20 2B 44 20 2B 45 20 2B 46 20 73 75 :88
6050 6D 20 3B 20 20 20 63 68 61 72 61 63 74 65 72 20 :F5
6060 20 20 20 3B 0D 0A 00 13 3A 2D 0D 0A 73 75 6D 20 :B8
6070 3A 20 00 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :FB
6080 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6090 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
60A0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
60B0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
60C0 E5 E5 E5 7F CF 42 20 05 86 FF B7 CF 42 CE 01 02 :82
60D0 86 01 F6 01 E6 ED C4 96 C3 F6 03 0D ED 42 FC 03 :A2
60E0 0E F0 03 0E 5C ED 44 B6 03 10 5F ED 46 B6 CF 42 :BE
60F0 26 03 B6 05 AD A7 48 8E E0 DC 0E DE 10 00 01 00 :C7
------------------------------------------------------------
Sum: 64 D3 93 67 69 1E E5 81 8F 5F 69 F2 23 E5 79 5E :46


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6100 00 0B 8E E0 F2 C6 06 A1 81 27 04 5A 26 F9 39 A6 :DC
6110 1F 39 2A 41 2F 42 2B 43 2D 44 3D 45 2C 46 00 00 :07
6120 BD D5 F1 FD CF 2D BD D5 F1 10 B3 CF 2D 10 25 F4 :E7
6130 7A FD CF 2F CE CF 31 6F C4 BD D5 BE C1 2E 26 0E :E9
6140 E6 80 A6 80 27 29 81 2C 10 26 F4 5F 20 13 30 1F :94
6150 BD D5 E0 10 27 F4 54 BD D5 93 27 13 4D 10 26 F4 :C7
6160 4A 6C C4 A6 C4 E7 C6 81 10 10 24 F4 3E 20 CA 4D :BF
6170 10 26 F4 37 6C C4 A6 C4 E7 C6 30 C4 FE CF 2D 1C :B2
6180 BF BD D6 D8 A1 01 27 0B 33 5F 11 B3 CF 2F 33 41 :C6
6190 25 EF 39 34 40 C6 01 E1 84 27 0C 5C BD D6 D8 A1 :88
61A0 85 27 F4 35 40 20 E1 EC E4 34 10 8E CB 00 83 00 :06
61B0 01 BD D8 27 CC 20 20 ED 81 ED 81 6F 84 8E CB 00 :F1
61C0 BD B8 23 35 50 7D 03 13 27 BE 7E D5 21 BD D5 F1 :8C
```

リスト2　スーパーモニタ，ローダープログラム　ダンプ・リスト

```
61D0 4D 10 26 F3 D6 39 8D F5 CE CF 43 F1 B8 0B 10 22 :CD
61E0 F3 C9 E7 47 A6 84 81 23 27 19 8D E1 C1 27 10 22 :80
61F0 F3 B9 E7 44 8D D7 5D 10 27 F3 B0 C1 20 10 22 F3 :78
-------------------------------------------------------------
Sum: AD D7 A8 D5 82 E4 F7 56 9E 07 E4 CA 7E 21 41 2E :15

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6200 AA 20 1A 30 01 8D C6 CB 08 C1 9F 10 22 F3 9C 34 :90
6210 04 54 54 E7 44 35 04 C4 03 58 58 58 5C 6F 46 C1 :B1
6220 11 25 04 6C 46 C0 10 E7 45 BD D5 F1 FD CF 4D BD :41
6230 D5 F8 10 B3 CF 4D 10 25 F3 71 10 83 80 00 10 24 :8C
6240 F3 69 FD CF 4F CC CB 00 ED 42 39 BD E1 B6 86 0A :5A
6250 A7 C4 BD E2 7B 8E CB 00 FE CF 4D FC CF 4F B3 CF :94
6260 4D 4D 27 09 5F BD D4 27 FF CF 4D 20 E5 5C BD D4 :EE
6270 27 39 BD E1 B6 86 09 A7 C4 BD D6 6B CE CB 00 BE :03
6280 CF 4D FC CF 4F B3 CF 4D 4D 27 0C 5F BD D4 27 BF :5B
6290 CF 4D BD E2 7B 20 E5 5C BD D4 27 86 10 F6 CF 47 :F1
62A0 C1 28 24 59 C6 05 34 0C 8E CF 43 1A 50 86 FD 1F :1D
62B0 8B A6 84 81 0A 27 05 BD FE 05 20 03 BD FE 08 35 :47
62C0 0C 4D 27 20 81 0A 27 35 81 0B 27 31 5A 27 2E 34 :4E
62D0 0C 86 FD 1F 8B 8E CF 43 BD FE 02 35 0C 81 0A 26 :88
62E0 C5 20 1A 39 6C 05 A6 05 81 11 25 F7 86 01 A7 05 :35
62F0 6C 06 A6 06 81 02 25 EB 6F 06 6C 04 39 80 09 8E :E6
-------------------------------------------------------------
Sum: D5 A5 65 DA CC 0A 0B 43 B5 D3 D5 83 5D D4 18 88 :8E

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6300 E2 F1 4A 27 06 E6 80 26 FC 20 F7 BD B8 23 7E D5 :D4
6310 21 21 21 20 44 72 69 76 65 20 6E 6F 74 20 72 65 :E5
6320 61 64 79 2E 07 0D 0A 00 21 21 20 44 69 73 6B 20 :97
6330 77 72 69 74 65 20 70 72 6F 74 65 63 74 65 64 2E :43
6340 07 0D 0A 00 21 21 20 48 61 72 64 20 65 72 72 6F :D7
6350 72 2E 07 0D 0A 00 21 21 20 43 52 43 20 65 72 72 :61
6360 6F 72 2E 07 0D 0A 00 21 21 20 44 44 20 4D 61 72 :57
6370 6B 20 64 65 74 65 63 74 65 64 2E 07 0D 0A 00 21 :3A
6380 21 20 44 69 73 6B 20 49 2F 4F 20 65 72 72 6F 72 :FD
6390 2E 07 0D 0A 00 21 21 20 49 6C 6C 65 67 61 6C 20 :88
63A0 74 72 61 63 6B 20 6E 75 6D 62 65 72 2E 07 0D 0A :0A
63B0 00 00 CC FF FF FD CF 73 B6 CF 14 B7 05 AC BD D5 :9C
63C0 E0 27 12 BD D5 93 34 06 27 06 BD D5 F8 FD CF 73 :6E
63D0 35 06 FD CF 71 FE CF 71 11 B3 CF 73 10 22 F1 61 :40
63E0 8E CB 00 86 24 A7 80 1F 30 BD D8 27 86 3A A7 80 :1C
63F0 BD D6 D8 BD D8 30 86 20 C6 0E 31 01 A7 80 5A 26 :83
-------------------------------------------------------------
Sum: 51 1C 55 06 81 26 8E 13 C1 7E AC E4 FC A8 6A E7 :D4

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6400 FB 7F CF 77 7F CF 75 7F CF 76 8D 5D FE CF 71 F6 :65
6410 CF 77 33 41 34 10 30 A4 5A 27 0C BD D6 D8 BD D8 :5F
6420 30 86 20 A7 80 20 F1 FF CF 71 35 10 CC 0D 0A ED :62
6430 81 6F 84 8E CB 00 BD B8 23 BD D8 56 27 97 39 34 :7B
6440 44 FE CF 71 F6 CF 77 33 C5 5C F7 CF 77 BD D6 D8 :BA
6450 35 C4 73 CF 75 27 05 7D CF 76 27 0D 7E E7 7C 73 :26
6460 CF 76 27 F8 7D CF 75 26 F3 8D D4 81 10 27 E3 81 :BB
6470 11 27 EC 85 80 10 26 00 44 85 F0 27 14 85 40 26 :3E
6480 10 CE E7 AD 80 12 8D 6B E6 C4 10 26 00 8D 7E E7 :CE
6490 7C CE E8 E1 34 02 84 0F 8D 59 E6 C4 A6 E0 84 70 :E6
64A0 26 09 86 40 C5 40 26 11 7E E7 7C 84 30 81 20 24 :8B
64B0 32 8B 10 44 44 A5 C4 27 EF 1F 89 20 5E 81 8D 26 :2E
64C0 0F 4F CE E7 A8 8D 2C C6 02 E5 C4 26 4E 7E E7 7C :3A
64D0 34 02 84 4F 85 40 27 02 80 30 CE E9 24 8D 14 A6 :C9
64E0 E0 84 30 C6 80 80 10 25 03 54 20 F9 E5 C4 26 2B :F9
64F0 7E E7 7C 4A 2A 1F 7D CF 75 26 0F 7D CF 76 27 14 :67
-------------------------------------------------------------
Sum: 59 36 5E 02 FA 39 45 1E C0 61 44 17 3A 4F DD E3 :4A

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6500 A6 C4 27 10 33 41 A6 C0 2A FC A6 C4 27 06 33 41 :AC
6510 A6 C0 2A FC 39 E6 C0 26 FC 20 D8 6D C4 10 27 02 :EF
6520 7B C4 FC 34 04 E6 C0 C4 03 EA E4 E7 E4 C6 06 A6 :EB
6530 C0 2B 05 A7 80 5A 20 F7 84 7F A7 80 5A 86 20 A7 :59
6540 80 5A 26 FB 35 04 C1 0F 26 01 39 C1 FF 26 05 CE :1D
6550 E7 90 20 07 C1 FE 26 28 CE E7 9C BD E4 1F 34 10 :00
6560 E6 C0 2B 04 E7 80 20 F8 C4 7F E7 80 C6 2C E7 80 :57
6570 44 25 04 35 10 20 02 32 62 6D C4 26 E1 30 1F 39 :28
6580 C1 03 10 27 00 94 C5 80 27 1A 86 23 A7 80 C5 02 :AC
6590 27 06 BD E4 1F 7E E7 14 BD E4 1F BD E7 14 BD E4 :7F
65A0 1F 7E D8 30 C5 40 27 07 CC 3C 24 ED 81 20 EF C5 :46
65B0 10 27 02 20 E3 C5 20 26 42 C5 08 27 04 C6 42 20 :A9
65C0 06 C5 04 27 0D C6 41 A6 82 81 20 27 FA 30 01 E7 :0C
65D0 80 39 C5 02 26 19 86 24 A7 80 BD E4 1F 1F 89 BD :B5
65E0 E4 1F 1E 89 F3 CF 71 FB CF 77 89 00 7E D8 27 86 :AA
65F0 24 A7 80 BD E4 1F 1F 89 1D 20 E9 BD E4 1F 85 80 :9E
-------------------------------------------------------------
Sum: BD B4 D5 EC AE ED 99 11 CE F0 A9 78 41 C3 A8 9C :9E

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6600 10 27 00 FC 85 10 27 08 34 02 86 5B A7 80 35 02 :6C
6610 1F 89 84 0F 48 CE E6 4C 6E D6 BD E4 1F 34 02 44 :01
6620 44 44 44 8D 10 86 2C A7 80 A6 E4 1F 89 E8 E0 C4 :00
6630 08 27 02 86 07 84 0F CE E6 2A 48 33 C6 A6 C0 A7 :7D
6640 80 A6 C0 81 20 27 02 A7 80 39 44 20 58 20 59 20 :65
6650 55 20 53 20 50 43 3F 3F 3F 3F 41 20 42 20 43 43 :C0
6660 44 50 3F 3F 3F 3F 3F 3F 3F 3F 3F 3F E6 8E E6 9B :FF
6670 E6 A5 E6 C4 E6 83 E6 D3 E6 D6 E7 7C E6 FE E7 1F :60
6680 E7 7C E6 D9 E7 5F E7 66 E7 7C E7 49 58 59 55 53 :A1
6690 34 04 59 59 59 59 C4 03 CE E6 6C E6 C5 86 2C ED :CD
66A0 81 35 84 8D EB C5 10 27 04 86 5D A7 80 39 C5 10 :CA
66B0 10 26 00 E8 8D DA 86 2B A7 80 39 8D D3 86 2B A7 :4E
66C0 80 A7 80 20 E0 C5 10 10 26 00 D1 86 2C A7 80 86 :E2
66D0 2D A7 80 34 04 59 59 59 59 C4 03 CE E6 6C A6 C5 :42
66E0 A7 80 35 84 34 04 CC 2C 2D ED 81 E7 80 35 04 8D :D8
66F0 E2 20 B2 86 42 8C 86 41 8C 86 44 A7 80 7E E6 83 :33
-------------------------------------------------------------
Sum: 5C 9F AC C7 8B 19 AA 52 84 D4 9C D1 FD 72 C1 20 :23

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6700 34 02 84 1F 85 10 27 04 84 0F 80 10 C6 2B 4D 2A :24
6710 03 C6 2D 40 E7 80 BD E7 14 35 04 7E E6 70 BD E4 :03
6720 1F 34 04 C6 2B 4D 2A 03 C6 2D 40 E7 80 35 04 8D :22
6730 03 7E E6 83 34 02 86 24 A7 80 35 02 7E D8 30 BD :6B
6740 E4 1F 34 04 1F 89 BD E4 1F 1E 89 34 04 C6 2B 4D :C0
6750 2A 0A E6 E4 50 E7 E4 89 00 40 C6 2D E7 80 8D D4 :9D
6760 35 02 8D D8 35 04 7E E6 83 C5 10 10 27 00 2D BD :B2
6770 E4 1F 8D C0 BD E4 1F BD E7 1C 86 5D A7 80 39 34 :47
6780 04 BD E5 CF 20 05 34 04 BD E5 B6 33 84 8E E7 8C :E2
6790 C6 04 BD D4 27 30 C4 35 04 7E E6 85 8E CB 16 CC :D3
67A0 3F 3F ED 81 A7 80 86 01 B7 CF 77 39 2C 50 43 52 :E1
67B0 43 C3 C1 C2 44 D0 D8 D9 D5 50 C3 00 43 C3 C1 C2 :BF
67C0 44 D0 D8 D9 D3 50 C3 00 02 42 53 D2 00 0F 4E 4F :C0
67D0 D0 00 0F 53 59 4E C3 00 00 00 01 4C 42 52 C1 00 :3E
67E0 01 4C 42 53 D2 00 00 0F 44 41 C1 00 82 4F 52 43 :6F
67F0 C3 00 00 82 41 4E 44 43 C3 00 0F 53 45 D8 00 03 :A0
-------------------------------------------------------------
Sum: A4 A3 48 0F 9D A8 F2 87 E4 35 D8 A7 ED 62 BE 6B :6C

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6800 45 58 C7 00 03 54 46 D2 00 02 42 52 C1 00 02 42 :6E
6810 52 CE 01 4C 42 52 CE 00 02 42 48 C9 01 4C 42 48 :FB
6820 C9 00 02 42 4C D3 01 4C 42 4C D3 00 02 42 43 C3 :24
6830 01 4C 42 43 C3 00 02 42 43 D3 01 4C 42 43 D3 00 :94
6840 02 42 4E C5 01 4C 42 4E C5 00 02 42 45 D1 01 4C :A0
6850 42 45 D1 00 02 42 56 C3 01 4C 42 56 C3 00 02 42 :A1
6860 56 D3 01 4C 42 56 D3 00 02 42 50 CC 01 4C 42 50 :20
6870 CC 00 02 42 4D C9 01 4C 42 4D C9 00 02 42 47 C5 :1B
6880 01 4C 42 47 C5 00 02 42 4C D4 01 4C 42 4C D4 00 :AE
6890 02 42 47 D4 01 4C 42 47 D4 00 02 42 4C C5 01 4C :AB
68A0 42 4C D4 00 20 4C 45 41 D8 00 20 4C 45 41 D9 00 :F7
68B0 20 4C 45 41 D3 00 20 4C 45 41 D5 00 FF 50 53 48 :76
68C0 D3 00 FF 50 55 4C D3 00 FE 50 53 48 D5 00 FE 50 :A2
68D0 55 4C D5 00 00 0F 52 54 D3 00 0F 41 42 D8 00 0F :77
68E0 52 54 C9 00 82 43 57 41 C9 00 0F 4D 55 CC 00 00 :12
68F0 0F 53 57 C9 0F 53 57 49 B2 0F 53 57 49 B3 00 00 :EB
-------------------------------------------------------------
Sum: B5 E5 C4 99 85 AF FF B1 1A B2 77 D2 98 29 E5 E3 :79

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6900 00 7C 4E 45 C7 00 00 00 7C 43 4F CD 00 7C 4C 53 :CC
6910 D2 00 00 7C 52 4F D2 00 7C 41 53 D2 00 7C 4C 53 :BE
6920 CC 00 7C 52 4F CC 00 7C 44 45 C3 00 00 7C 49 4E :90
6930 C3 00 7C 54 53 D4 00 70 4A 4D D0 00 7C 43 4C D2 :6E
6940 00 00 00 00 F2 53 55 42 C1 00 F2 43 4D 50 C1 00 :30
6950 F2 53 42 43 C1 00 F1 53 55 42 C4 F1 43 4D 50 C4 :BF
6960 F1 43 4D 50 D5 00 F2 41 4E 44 C1 00 F2 42 49 54 :FD
6970 C1 00 F2 4C 44 C1 00 70 53 54 C1 00 F2 45 4F 52 :B4
6980 C1 00 F2 41 44 43 C1 00 F2 4F 52 C1 00 F2 41 44 :07
6990 44 C1 00 F1 43 4D 50 D8 F1 43 4D 50 D9 F1 43 4D :D9
69A0 50 D3 00 70 4A 53 D2 00 F1 4C 44 D8 F1 4C 44 D9 :B5
69B0 00 70 53 54 D8 70 53 54 D9 00 F2 53 55 42 C2 00 :7D
69C0 F2 43 4D 50 C2 00 F2 53 42 43 C2 00 F1 41 44 44 :DA
69D0 C4 00 F2 41 4E 44 C2 00 F2 42 49 54 C2 00 F2 4C :1C
69E0 44 C2 00 70 53 54 C2 00 F2 45 4F 52 C2 00 F2 41 :AC
69F0 44 43 C2 00 F2 4F 52 C2 00 F2 41 44 44 C2 00 F1 :0C
-------------------------------------------------------------
Sum: 98 5E 0D 3D 85 3D 08 73 10 8A DD F9 C8 4F 88 5C :E8

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6A00 4C 44 C4 00 70 53 54 C4 00 F1 4C 44 D5 F1 4C 44 :06
6A10 D3 00 70 53 54 D5 70 53 54 D3 00 BD D5 E0 26 2D :6E
6A20 5D 10 26 EB 86 CE EA F8 8E CB 00 CC 30 20 ED 81 :97
6A30 EC C1 10 83 FF FF 27 14 BD D8 27 CC 0D 0A ED 81 :86
6A40 6F 84 8E CB 00 BD B8 23 6C 81 20 E4 39 BD D5 F1 :91
6A50 10 83 80 00 10 24 EB 53 FD CF 82 A6 80 80 52 10 :DB
6A60 25 EB 48 81 01 10 22 EB 42 CE EA F8 C6 0A AE C1 :28
6A70 8C FF FF 27 11 BC CF 82 27 1A 5A 26 F1 4D 27 CC :C1
6A80 8E EA E0 7E B8 23 4D 27 0A FC CF 82 ED 5E CC FF :92
6A90 FF ED C4 39 4D 26 FC EC C1 ED 5C 10 83 FF FF 26 :05
6AA0 F6 39 8E EA F8 10 8E CF 78 EC 84 10 83 FF FF 27 :AC
6AB0 F0 11 A3 84 27 0D E6 A0 A6 91 81 3F 26 EB E7 98 :69
6AC0 FE 20 E6 E6 A0 EE 81 A6 C4 81 3F 26 02 E7 C4 EE :E4
6AD0 81 EF 1C E6 A0 E7 3E 11 83 FF FF 27 C4 A6 C4 81 :9F
6AE0 3F 26 EC E7 C4 20 E8 8E EA F8 10 8E CF 78 EE 81 :C8
6AF0 11 83 FF FF 27 AB A6 C4 A7 A0 86 3F A7 C4 20 EE :53
-------------------------------------------------------------
Sum: DA DF 81 0B BA A8 73 91 32 1D 5D 3C AC 9F 8F C3 :30

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6B00 21 21 20 42 72 65 61 6B 20 70 6F 69 6E 74 20 66 :17
6B10 75 6C 6C 2E 07 0D 0A 00 FF FF E5 E5 E5 E5 E5 E5 :F5
6B20 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6B30 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6B40 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6B50 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6B60 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6B70 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6B80 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6B90 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6BA0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6BB0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6BC0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6BD0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6BE0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
6BF0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
-------------------------------------------------------------
Sum: 1C 13 12 F6 FF F8 F1 F1 A5 F5 DA D4 D9 DF 8B D1 :6C
```

リスト3　フォーマット・プログラム

```
1000 CLEAR 300,&H3000
1010 LOADM 'format.M'
1020 INPUT 'FORMAT DISK DRIVE (0,1)';DRV : IF DRV<0 OR DRV>1 GOTO 1020
1030 POKE &H3003,DRV
1040 PRINT 'SET DISK IN DRIVE ';DRV
1050 PRINT 'IF READY THEN PRESS ANY KEY ' : A$=INPUT$(1)
1060 EXEC &H3000
1070 IF PEEK(&H3002)<>0 THEN PRINT 'DISK I/O ERROR !!!' : BEEP : GOTO 1040
1080 PRINT 'COMPLETE.' : PRINT
1090 END
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3000 20 05 00 00 00 00 00 34 08 86 FD 1F 8B 7F 30 02 :3F
3010 7F 30 04 8E 31 16 86 08 A7 84 B6 30 03 A7 07 BD :95
3020 FE 02 26 16 0D 18 2B 12 8D 3C 26 0E B6 30 04 81 :06
3030 27 27 0A 7C 30 04 8D 07 27 EA 73 30 02 35 88 86 :95
3040 5A D6 18 54 25 FB 97 18 8D 13 D6 1F C5 40 27 FA :26
3050 10 8E 0A 00 31 3F 26 FC D6 18 C5 98 39 10 8E 00 :5C
3060 7B 31 3F 26 FC 39 7F 30 05 0F 1C 8D 07 26 F6 7C :51
3070 30 05 0C 1C 7F 30 06 CE 31 06 10 8E 31 16 CC 4E :16
3080 20 8D 5D CC 00 0C 8D 58 CC F5 03 8D 53 86 FE A7 :96
3090 A0 B6 30 04 A7 A0 B6 30 05 A7 A0 B6 30 06 A6 C6 :5B
30A0 7C 30 06 A7 A0 86 01 A7 A0 86 F7 A7 A0 CC 4E 16 :BB
30B0 8D 2E CC 00 0C 8D 29 CC F5 03 8D 24 86 FB A7 A0 :86
30C0 CC E5 00 8D 1B 86 F7 A7 A0 CC 4E 36 8D 12 F6 30 :32
30D0 06 C1 10 26 AE CC 4E 00 8D 06 8D 04 8D 02 20 06 :9E
30E0 A7 A0 5A 26 FB 39 10 8E 31 16 86 F4 97 18 A6 A0 :4F
30F0 D6 1F 2A 04 97 1B 20 F6 58 2A F5 D6 18 C5 C4 26 :FF
---------------------------------------------------------
Sum: F1 FE 94 0A ED 3A 62 8D 18 A7 90 71 EE 5B 53 A9 :A8
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3100 04 BD 30 5D 4F 39 01 09 04 0C 07 0F 02 0A 05 0D :24
3110 08 10 03 0B 06 0E E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :2C
3120 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3130 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3140 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3150 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3160 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3170 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3180 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3190 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
31A0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
31B0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
31C0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
31D0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
31E0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
31F0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
---------------------------------------------------------
Sum: 92 53 B9 EE DB CD 6C 74 6F 77 72 7A 6D 75 70 78 :B0
```

リスト4　スタート・アップ，フリーエリア拡張プログラム

スタート・アップ・プログラム例　Sample program of auto start　(FILE NAME : startup)

```
10 CLEAR ,&H5000
20 LOADM 'MONITOR',,R
30 CLEAR ,&H8000
40 EXEC &H8632
```

フリーエリア拡張プログラム　Sample program of BASIC free area expander

```
10 DEF FNA(A)=PEEK(A)*256+PEEK(A+1)
20 NOWTOP=FNA(&H33)
30 IF FNA(NOWTOP-3)<>&HB806 THEN PRINT 'Already expanded.':END
40 EXTOP=NOWTOP-&H5D
50 POKE &H33,EXTOP¥256:POKE &H34,EXTOP MOD 256
60 POKE EXTOP-1,0
70 POKE &H01FA,&H80:POKE &H01FB,&H1D
80 POKE &H01FC,&H80:POKE &H01FD,&H5A
90 POKE &H01FF,&H80:POKE &H0200,&H1D
100 POKE &H0201,&H80:POKE &H0202,&H77
110 EXEC &H8632
```

RANDOM BOX

“6809デバッガ”のFM-7への移植

■COMPAC S

6809デバッガはマシン語入力，マシン語ダンプ，逆アセンブルなどに便利なユーティリィティです．FM-7でもそのまま動くのですが，テン・キーの配置が変わっているので，Mコマンドでマシン語入力するとき，本来の便利な機能が活かされません．

FM-7のテン・キーに合わせて，A，B，C，D，E，Fの入力を図1のように変更し，ついでに，画面の初期状態を80字表示にしました．アドレスのインクリメント/デクリメントは，それぞれ[SPACE]，[←]であり，変更ありません．その他のキー入力で，マシン語入力が終わります．また，アドレスの最下位が＄Fのあとは，そのアドレスまでの16バイトのチェック・サムを表示することも，変わりありません．

◘参考文献

1）笠作貴弥，“6809デバッガ”，FM-8活用研究，工学社

図1　Mコマンドのキー配置

A	B	C	D
7	8	9	E
4	5	6	F
1	2	3	↵
0			↵

リスト変更箇所

add	old	new
69A2	28	50
6C7A	2E	2A
6C7E	0D	2F
6C82	2D	2B
6C86	2B	2D
6C8A	2C	3D
6C8E	1C	2C

FM-7の裏RAMにテキストを格納

エディタ・アセンブラ⇔BASIC コミュニケーションプログラム

■吉川尚之

このプログラムは，馬場二郎氏のエディタ・アセンブラ 2.0(I/O '83年５月号，FM-7/8活用研究)をALL RAM MODEで使うためのプログラムです．

FM-8用は'83年７月号（RANDOM BOX）で発売されましたが，テキストがメインRAM上にあるため，一度暴走するとテキストが破壊されてしまう恐れがあります．そこで，テキストも裏RAMで使えるように自作しました．

特　　徴

❶6809ではあたりまえですが，完全リロケータブルです．
❷テキスト，アセンブラ本体共に裏RAM上にあるため，効率よくデバッグが行なえる．
❸長いプログラムを一度でアセンブラできます．

入力・実行方法

CLEAR 100, &H1C3FとしてBASICエリアを確保した後，&H1C40からプログラムを入力します．入力したらすぐにセーブしてください．

セーブ方法は，**SAVEM"ファイル名", &H1C40, &H1FFF, &H1C40**です．

実行方法は，**CLEAR 100, &H1C3F**の後，**LOADM"ファイル名",, R**で走ります．スタート番地は&H1C40です．

なお，メモリ・マップは図１のとおりです．

コマンド説明

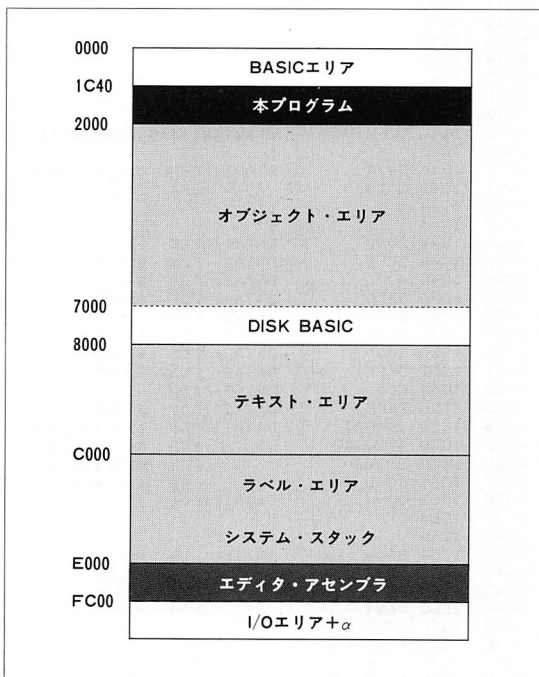
コマンドは６つ用意されています．

[1]：エディタ・アセンブラをメインRAMから裏RAMへ転送します．

実行時にはメインRAMの&H4000から&5BFFにアセンブラ本体がなければなりません．転送後はBASICエリア保護のため，自動的にテキストの１行目に"**ORG $2000**"とセットされます．また，転送は裏RAMをすべてクリアした後で行なうため，テキストが裏RAMにあってもクリアされ

コマンド一覧表

コマンド	機　　能
1	アセンブラ転送
2	テキスト転送（メイン→裏）
3	テキスト転送（裏→メイン）
A	アセンブラへジャンプ
B	BASICへ戻る
H	コマンド表示

メモリ・マップ

てしまいます．そこで誤操作防止のために，一度このコマンドを実行すると，メインRAM上のアセンブラ 本体の一部を変更して，転送できなくしています．

もう一度転送する場合には，一度BASICへ戻り，アセンブラをロードする必要があります．

[2]：テキストをメインRAMから裏RAMへ転送します．

エラーチェックを行なっていないので，必ずメインRAMの&H2000からにテキストをロードした後に実行してください．

3：テキストを裏RAMからメインRAMへ転送します．
　転送後アドレスが表示されるので，それに従ってセーブしてください．
A：エディタ・アセンブラ（裏RAM上）へジャンプします．裏RAM上にアセンブラが転送されてる必要があります．
B：BASICへ戻ります．
H：コマンドを表示します．

使用上の注意

　FM-7の場合は，自動的にROMモードとALL RAMモードの切り換えを行なっていますが，FM-8の場合は画面の指示に従って，本体裏のディップ・スイッチ９番を切り換えてください．また，オブジェクトのテストなどで暴走した場合，リセット・ボタンを押し，**CLEAR**実行後，本プログラムを走らせると，裏RAM上でアセンブラ本体，テキストともに無傷で残っています（けっして電源を切らないこと）．なお，プログラムやテキストのロード，セーブ，オブジェクトのテストはBASICへ戻って行なってください．

最後に

　馬場二郎氏のエディタ・アセンブラはとてもよくできたプログラムなので，本プログラムを使って，その能力を充分活用していただければ幸いかと思います．なお，私は**リスト2**のプログラムで，ロードのてまを省いています．

参考文献

1)FM-8活用研究
2)FM-7/8活用研究
3)庄野温夫：〝エディタ・アセンブラV2.0をALL RAMで使う〞，I/O'83年７月号
4)FM-8機械語教本F-BIOSの使い方，(株)コマス

リスト１　コミュニケーション・プログラム

```
0001  0000                 ;**********************************
0002  0000                 ;**                              **
0003  0000                 ;**  ｴﾃﾞｨﾀ･ｱｾﾝﾌﾞﾗ V.2.0 - BASIC  **
0004  0000                 ;**                              **
0005  0000                 ;**   ｺﾐﾆｭｹｰｼｮﾝ  ﾌﾟﾛｸﾞﾗﾑ          **
0006  0000                 ;**                              **
0007  0000                 ;**   83/10/15   by N.YOSHIKAWA  **
0008  0000                 ;**                              **
0009  0000                 ;**********************************
0010  0000                 ;
0011  0000                 ;
0012  0000                 ;
0013  1C40                         ORG   $1C40
0014  1C40                 ;
0015  1C40                 HLT:    EQU   $FD05
0016  1C40                 ;
0017  1C40                 ;
0018  1C40                 ;**********************************
0019  1C40                 ;
0020  1C40 347F            START:  PSHS A,B,X,Y,U,DP,CC
0021  1C42 2020                    BRA   INIT
0022  1C44                 ;
0023  1C44                 ;
0024  1C44 2000            DATA1:  FDB   $2000           ;TEXT    ｾﾝﾄｳ ｱﾄﾞﾚｽ (ｵﾓﾃ)
0025  1C46 5FFF            DATA2:  FDB   $5FFF           ;TEXT    ｴﾝﾄﾞ ｱﾄﾞﾚｽ (ｵﾓﾃ)
0026  1C48 4000            DATA3:  FDB   $4000           ;ｱｾﾝﾌﾞﾗ ｾﾝﾄｳ ｱﾄﾞﾚｽ (ｵﾓﾃ)
0027  1C4A 5BFF            DATA4:  FDB   $5BFF           ;ｱｾﾝﾌﾞﾗ ｴﾝﾄﾞ ｱﾄﾞﾚｽ (ｵﾓﾃ)
0028  1C4C 8000            DATA5:  FDB   $8000           ;TEXT    ｾﾝﾄｳ ｱﾄﾞﾚｽ (ｳﾗ)
0029  1C4E BFFF            DATA6:  FDB   $BFFF           ;TEXT    ｴﾝﾄﾞ ｱﾄﾞﾚｽ (ｳﾗ)
0030  1C50 C000            DATA7:  FDB   $C000           ;ﾗﾍﾞﾙ    ｾﾝﾄｳ ｱﾄﾞﾚｽ (ｳﾗ)
0031  1C52 DFFF            DATA8:  FDB   $DFFF           ;ﾗﾍﾞﾙ    ｴﾝﾄﾞ ｱﾄﾞﾚｽ (ｳﾗ)
0032  1C54 E000            DATA9:  FDB   $E000           ;ｱｾﾝﾌﾞﾗ ｾﾝﾄｳ ｱﾄﾞﾚｽ (ｳﾗ)
0033  1C56 FBFF            DATA10:FDB    $FBFF           ;ｱｾﾝﾌﾞﾗ ｴﾝﾄﾞ ｱﾄﾞﾚｽ (ｳﾗ)
0034  1C58 204F52          DATA11:FCB    $20,$4F,$52     ;ORG $2000 DATA
0035  1C5B 472024                  FCB   $47,$20,$24
0036  1C5E 323030                  FCB   $32,$30,$30
0037  1C61 300D                    FCB   $30,$0D
0038  1C63 00              DATA12:FCB    $00
0039  1C64                 ;
0040  1C64                 ;
0041  1C64 8604            INIT:   LDA   #$04            ;*****  INIT  *****
0042  1C66 170259                  LBSR COLOR            ;COLOR 4
0043  1C69 B7FD0F                  STA   $FD0F
0044  1C6C 8634                    LDA   #$34
0045  1C6E A79DFFE2                STA   [DATA9,PC]      ;FM-7 OR FM-8
0046  1C72 A19DFFDE                CMPA [DATA9,PC]
0047  1C76 271A                    BEQ   INI1
0048  1C78 308D027C                LEAX PD1,PC
0049  1C7C 17023A                  LBSR PRINT2           ;PRINT "DIP ｽｲｯﾁ ﾀﾞﾊﾞﾝ "
0050  1C7F 308D028C                LEAX PD3,PC           ;PRINT "OFF"
0051  1C83 170233                  LBSR PRINT2
0052  1C86 8634                    LDA   #$34
0053  1C88 AE8DFFC8                LDX   DATA9,PC
0054  1C8C A784            DIPOFF:STA    ,X              ;DIP OFF ?
0055  1C8E A184                    CMPA ,X
0056  1C90 26FA                    BNE   DIPOFF
0057  1C92 308D02B9        INI1:   LEAX PD8,PC           ;PRINT  ｺﾏﾝﾄﾞ ｲﾁﾗﾝ
0058  1C96 170220                  LBSR PRINT2
0059  1C99 308D02A7        MAIN:   LEAX PD7,PC           ;****  MAIN  *****
0060  1C9D 170219                  LBSR PRINT2
0061  1CA0 170236          MAI1:   LBSR INPUT
```

リスト 1

```
0062  1CA3 A78DFFBC           STA   DATA12,PC
0063  1CA7 8131               CMPA  #$31          ;A<"1" THEN MAI1
0064  1CA9 25F5               BCS   MAI1
0065  1CAB 8148               CMPA  #$48          ;A>"H" THEN MAI1
0066  1CAD 22F1               BHI   MAI1
0067  1CAF 1701F5             LBSR  PRINT1
0068  1CB2 170224      MAI2:  LBSR  INPUT
0069  1CB5 8108               CMPA  #$08          ;A=BS THEN MAI1
0070  1CB7 260A               BNE   MAI3
0071  1CB9 1701EB             LBSR  PRINT1
0072  1CBC 8605               LDA   #$05
0073  1CBE 1701E6             LBSR  PRINT1
0074  1CC1 20DD               BRA   MAI1
0075  1CC3 810D        MAI3:  CMPA  #$0D          ;A<>CR THEN MAI2
0076  1CC5 26EB               BNE   MAI2
0077  1CC7             ;
0078  1CC7 1701DD             LBSR  PRINT1
0079  1CCA 860A               LDA   #$0A
0080  1CCC 1701D8             LBSR  PRINT1
0081  1CCF A68DFF90           LDA   DATA12,PC
0082  1CD3 8131               CMPA  #$31
0083  1CD5 2605               BNE   MA1
0084  1CD7 170034             LBSR  TENSOU        ;A="1" THEN GOS. TENSOU
0085  1CDA 20BD               BRA   MAIN
0086  1CDC 8132        MA1:   CMPA  #$32
0087  1CDE 2605               BNE   MA2
0088  1CE0 170089             LBSR  TEXT1         ;A="2" THEN GOS. TEXT1
0089  1CE3 20B4               BRA   MAIN          ;              (RAM -> ROM)
0090  1CE5 8133        MA2:   CMPA  #$33
0091  1CE7 2605               BNE   MA3
0092  1CE9 17009F             LBSR  TEXT2         ;A="3" THEN GOS. TEXT2
0093  1CEC 20AB               BRA   MAIN          ;              (ROM -> RAM)
0094  1CEE 8141        MA3:   CMPA  #$41
0095  1CF0 2605               BNE   MA4
0096  1CF2 170102             LBSR  ASSEM         ;A="A" THEN GOS. ASSEM
0097  1CF5 20A2               BRA   MAIN          ;                (アセンブ゛ラ)
0098  1CF7 8142        MA4:   CMPA  #$42
0099  1CF9 2603               BNE   MA5
0100  1CFB 16012F             LBRA  BASIC         ;A="B" THEN BASIC
0101  1CFE 8148        MA5:   CMPA  #$48
0102  1D00 2605               BNE   MA6
0103  1D02 170158             LBSR  HELP          ;A="H" THEN HELP
0104  1D05 2092               BRA   MAIN          ;        (コマント゛ イチラン)
0105  1D07 8607        MA6:   LDA   #$07
0106  1D09 17019B             LBSR  PRINT1
0107  1D0C 208B               BRA   MAIN
0108  1D0E             ;
0109  1D0E             ;
0110  1D0E EC9DFF36    TENSOU:LDD   [DATA3,PC]    ;****   1 (アセンブ゛ラ テソウ)   ****
0111  1D12 1083347F           CMPD  #$347F        ;エラー チェック
0112  1D16 2710               BEQ   TENS01
0113  1D18 308D01F9           LEAX  PD4,PC
0114  1D1C 17019A             LBSR  PRINT2
0115  1D1F 308D01FE           LEAX  PD6,PC        ;PRINT"メイン RAM ニ ASSMBLER"
0116  1D23 170193             LBSR  PRINT2        ;PRINT"フ゜ログ゛ラム カ゛ アリマセン"
0117  1D26 2043               BRA   TENS03
0118  1D28 AE8DFF20    TENS01:LDX   DATA5,PC      ;ウラ RAM CLEAR
0119  1D2C EE8DFF26           LDU   DATA10,PC
0120  1D30 170145             LBSR  CLEAR
0121  1D33 AE8DFF11           LDX   DATA3,PC      ;フ゜ログ゛ラム テンソウ
0122  1D37 EE8DFF19           LDU   DATA9,PC
0123  1D3B 10AE8DFF0A         LDY   DATA4,PC
0124  1D40 17012A             LBSR  PTENSU
0125  1D43 AE8DFF0D           LDX   DATA9,PC      ;TEXT・エリア セントウアト゛レス
0126  1D47 EC8DFF01           LDD   DATA5,PC      ;          &
0127  1D4B ED09               STD   9,X           ;ラヘ゛ル・エリア セントウアト゛レス SET
0128  1D4D EC8DFEFF           LDD   DATA7,PC
0129  1D51 ED07               STD   7,X
0130  1D53 ED9DFEF1           STD   [DATA3,PC]
0131  1D57 308DFEFD           LEAX  DATA11,PC     ;ORG $1000 ヲ セット
0132  1D5B EE8DFEED           LDU   DATA5,PC
0133  1D5F C60B               LDB   #$0B
0134  1D61 A680        TENS02:LDA   ,X+
0135  1D63 A7C0               STA   ,U+
0136  1D65 5A                 DECB
0137  1D66 26F9               BNE   TENS02
0138  1D68 1700FA             LBSR  PP1           ;PRINT"テンソウ シュウリョウ !!"
0139  1D6B 39          TENS03:RTS
0140  1D6C             ;
0141  1D6C             ;
0142  1D6C AE8DFEDC    TEXT1: LDX   DATA5,PC      ;*** TEXTテンソウ1 (オモテ→ウラ) ***
0143  1D70 EE8DFEDE           LDU   DATA8,PC      ;ウラ TEXT & ラヘ゛ル リョウイキ CLEAR
0144  1D74 170101             LBSR  CLEAR
0145  1D77 AE8DFEC9           LDX   DATA1,PC      ;TEXT テンソウ
0146  1D7B EE8DFECD           LDU   DATA5,PC
0147  1D7F 10AE8DFEC2         LDY   DATA2,PC
0148  1D84 1700E6             LBSR  PTENSU
0149  1D87 1700DB             LBSR  PP1           ;PRINT"テンソウ シュウリョウ !!"
0150  1D8A 39                 RTS
0151  1D8B             ;
0152  1D8B             ;
```

リスト1

```
0153  1D8B AE8DFEBD     TEXT2: LDX  DATA5,PC        ;*** TEXTテンソウ2 (ウラ▲オモテ) ***
0154  1D8F EE8DFEB1            LDU  DATA1,PC
0155  1D93 10AE8DFEB6          LDY  DATA6,PC        ;TEXT テンソウ
0156  1D98 1700D2              LBSR PTENSU
0157  1D9B 338D0245            LEAU PD11,PC
0158  1D9F EC8DFEA1            LDD  DATA1,PC
0159  1DA3 8D2E                BSR  TEX
0160  1DA5 AE8DFEA3            LDX  DATA5,PC
0161  1DA9 1F10                TFR  X,D
0162  1DAB EE80         TEXT3: LDU  ,X+
0163  1DAD C30001              ADDD #$0001
0164  1DB0 11830000            CMPU #$0000
0165  1DB4 26F5                BNE  TEXT3
0166  1DB6 A38DFE92            SUBD DATA5,PC
0167  1DBA E38DFE86            ADDD DATA1,PC
0168  1DBE 338D022B            LEAU PD12,PC
0169  1DC2 8D0F                BSR  TEX
0170  1DC4 308D0203            LEAX PD9,PC
0171  1DC8 1700EE              LBSR PRINT2
0172  1DCB 308D020D            LEAX PD10,PC
0173  1DCF 1700E7              LBSR PRINT2
0174  1DD2 39                  RTS
0175  1DD3              ;
0176  1DD3 1F01         TEX:   TFR  D,X
0177  1DD5 8D0D                BSR  TE1
0178  1DD7 1F10                TFR  X,D
0179  1DD9 8D0D                BSR  TE2
0180  1DDB 1F98                TFR  B,A
0181  1DDD 8D05                BSR  TE1
0182  1DDF 1F98                TFR  B,A
0183  1DE1 8D05                BSR  TE2
0184  1DE3 39                  RTS
0185  1DE4              ;
0186  1DE4 44           TE1:   LSRA
0187  1DE5 44                  LSRA
0188  1DE6 44                  LSRA
0189  1DE7 44                  LSRA
0190  1DE8 840F         TE2:   ANDA #$0F
0191  1DEA 8109                CMPA #$09
0192  1DEC 2204                BHI  TE3
0193  1DEE 8B30                ADDA #$30
0194  1DF0 2002                BRA  TE4
0195  1DF2 8B37         TE3:   ADDA #$37
0196  1DF4 A7C0         TE4:   STA  ,U+
0197  1DF6 39                  RTS
0198  1DF7              ;
0199  1DF7              ;
0200  1DF7 EC9DFE59     ASSEM: LDD  [DATA9,PC]      ;***** アセンブ゛ラ ヘ JUMP *****
0201  1DFB 1083347F            CMPD #$347F
0202  1DFF 2710                BEQ  ASSE1           ;エラー チェック
0203  1E01 308D0116            LEAX PD5,PC
0204  1E05 1700B1              LBSR PRINT2
0205  1E08 308D0115            LEAX PD6,PC          ;PRINT"ウラ RAM ニ ASSEMBLER"
0206  1E0C 1700AA              LBSR PRINT2          ;PRINT"プ゜ログ゛ラム ガ゛ アリマセン"
0207  1E0F 201B                BRA  ASSE2
0208  1E11 8607         ASSE1: LDA  #$07            ;COLOR 7
0209  1E13 1700AC              LBSR COLOR
0210  1E16 1F40                TFR  S,D             ;アセンブ゛ラ ヘ JUMP
0211  1E18 10EE8DFE35          LDS  DATA8,PC
0212  1E1D 3406                PSHS A,B
0213  1E1F AD9DFE31            JSR  [DATA9,PC]
0214  1E23 3506                PULS A,B
0215  1E25 1F04                TFR  D,S
0216  1E27 8604                LDA  #$04
0217  1E29 170096              LBSR COLOR
0218  1E2C 39           ASSE2: RTS
0219  1E2D              ;
0220  1E2D              ;
0221  1E2D 8607         BASIC: LDA  #$07            ;*****  BASIC ヘ モト゛ル *****
0222  1E2F 170090              LBSR COLOR           ;COLOR 7
0223  1E32 B6FD0F              LDA  $FD0F
0224  1E35 8634                LDA  #$34
0225  1E37 A79DFE19            STA  [DATA9,PC]      ;FM-7 OR FM-8
0226  1E3B A19DFE15            CMPA [DATA9,PC]
0227  1E3F 261A                BNE  BASI2
0228  1E41 308D00B3            LEAX PD1,PC
0229  1E45 170071              LBSR PRINT2          ;PRINT"DIP スイッチ 9バ゛ン "
0230  1E48 308D00BE            LEAX PD2,PC          ;PRINT"ON"
0231  1E4C 17006A              LBSR PRINT2
0232  1E4F 8634                LDA  #$34
0233  1E51 AE8DFDFF            LDX  DATA9,PC
0234  1E55 A784         BASI1: STA  ,X              ;DIP ON ?
0235  1E57 A184                CMPA ,X
0236  1E59 27FA                BEQ  BASI1
0237  1E5B 35FF         BASI2: PULS A,B,X,Y,U,DP,CC,PC
0238  1E5D              ;
0239  1E5D              ;
0240  1E5D 308D00EE     HELP:  LEAX PD8,PC          ;**** コマンド゛ イチラン PRINT ****
0241  1E61 170055              LBSR PRINT2
0242  1E64 39                  RTS
0243  1E65              ;
```

リスト1

```
0244  1E65                  ;
0245  1E65                  ;**********  ｻﾌﾞﾙｰﾁﾝ  ************
0246  1E65                  ;
0247  1E65 308D0162  PP1:   LEAX PD9,PC         ;PRINT"ﾃﾝｿｳ ｼｭｳﾘｮｳ" SUB
0248  1E69 17004D           LBSR PRINT2
0249  1E6C 39               RTS
0250  1E6D                  ;
0251  1E6D                  ;
0252  1E6D EC81      PTENSU:LDD  ,X++           ;ﾃﾝｿｳ SUB
0253  1E6F EDC1             STD  ,U++
0254  1E71 3420             PSHS Y
0255  1E73 ACE1             CMPX ,S++
0256  1E75 25F6             BCS  PTENSU
0257  1E77 39               RTS
0258  1E78                  ;
0259  1E78                  ;
0260  1E78 4F        CLEAR: CLRA                ;CLEAR SUB
0261  1E79 5F               CLRB
0262  1E7A ED81      CLEA1: STD  ,X++
0263  1E7C 3440             PSHS U
0264  1E7E ACE1             CMPX ,S++
0265  1E80 25F8             BCS  CLEA1
0266  1E82 39               RTS
0267  1E83                  ;
0268  1E83                  ;
0269  1E83 3402      SUBHLT:PSHS A              ;SUBCPU HLT SUB
0270  1E85 B6FD05    SUBHL1:LDA  HLT
0271  1E88 2BFB             BMI  SUBHL1
0272  1E8A 1A50             ORCC #$50
0273  1E8C 8680             LDA  #$80
0274  1E8E B7FD05           STA  HLT
0275  1E91 B6FD05    SUBHL2:LDA  HLT
0276  1E94 2AFB             BPL  SUBHL2
0277  1E96 B6FC82           LDA  $FC82
0278  1E99 2704             BEQ  SUBHL3
0279  1E9B 8D04             BSR  HLTCLR
0280  1E9D 20E6             BRA  SUBHL1
0281  1E9F 3582      SUBHL3:PULS A,PC
0282  1EA1                  ;
0283  1EA1                  ;
0284  1EA1 7FFD05    HLTCLR:CLR  HLT            ;SUBCPU HLT CLEAR SUB
0285  1EA4 1CAF             ANDC #$AF
0286  1EA6 39               RTS
0287  1EA7                  ;
0288  1EA7                  ;
0289  1EA7 8DDA      PRINT1:BSR  SUBHLT         ;1ﾓｼﾞ PRINT SUB
0290  1EA9 C603             LDB  #$03
0291  1EAB F7FC82           STB  $FC82
0292  1EAE C601             LDB  #$01
0293  1EB0 F7FC83           STB  $FC83
0294  1EB3 B7FC84           STA  $FC84
0295  1EB6 8DE9             BSR  HLTCLR
0296  1EB8 39               RTS
0297  1EB9                  ;
0298  1EB9                  ;
0299  1EB9 A680      PRINT2:LDA  ,X+            ;ﾚﾝｿﾞｸ PRINT SUB
0300  1EBB 2704             BEQ  PRIN2
0301  1EBD 8DE8             BSR  PRINT1
0302  1EBF 20F8             BRA  PRINT2
0303  1EC1 39        PRIN2: RTS
0304  1EC2                  ;
0305  1EC2                  ;
0306  1EC2 8DBF      COLOR: BSR  SUBHLT         ;COLOR SUB
0307  1EC4 C603             LDB  #$03
0308  1EC6 F7FC82           STB  $FC82
0309  1EC9 C602             LDB  #$02
0310  1ECB F7FC83           STB  $FC83
0311  1ECE C611             LDB  #$11
0312  1ED0 F7FC84           STB  $FC84
0313  1ED3 B7FC85           STA  $FC85
0314  1ED6 8DC9             BSR  HLTCLR
0315  1ED8 39               RTS
0316  1ED9                  ;
0317  1ED9                  ;
0318  1ED9 8DA8      INPUT: BSR  SUBHLT         ;1ﾓｼﾞ INPUT SUB
0319  1EDB 8629             LDA  #$29
0320  1EDD B7FC82           STA  $FC82
0321  1EE0 8603             LDA  #$03
0322  1EE2 B7FC83           STA  $FC83
0323  1EE5 8DBA             BSR  HLTCLR
0324  1EE7 8D9A             BSR  SUBHLT
0325  1EE9 B6FC83           LDA  $FC83
0326  1EEC C680             LDB  #$80
0327  1EEE F7FC80           STB  $FC80
0328  1EF1 8DAE             BSR  HLTCLR
0329  1EF3 8100             CMPA #$00
0330  1EF5 27E2             BEQ  INPUT
0331  1EF7 39               RTS
0332  1EF8                  ;
0333  1EF8                  ;
0334  1EF8                  ;
```

リスト１

```
0335  1EF8                ;******  PRINT DATA ｴﾘｱ  ******
0336  1EF8                ;
0337  1EF8 070D0A         PD1:    FCB  $07,$0D,$0A
0338  1EFB 44495020               FCC  /DIP /
0339  1EFF BDB2AFC1               FCC  /ｽｲｯﾁ/
0340  1F03 2039CADE               FCC  / 9ﾊﾞ/
0341  1F07 DD20                   FCC  /ﾝ /
0342  1F09 00                     FCB  $00
0343  1F0A 4F4E           PD2:    FCC  /ON/
0344  1F0C 0D0A00                 FCB  $0D,$0A,$00
0345  1F0F 4F4646         PD3:    FCC  /OFF/
0346  1F12 0D0A00                 FCB  $0D,$0A,$00
0347  1F15 07             PD4:    FCB  $07
0348  1F16 D2B2DD20               FCC  /ﾒｲﾝ /
0349  1F1A 00                     FCB  $00
0350  1F1B 07             PD5:    FCB  $07
0351  1F1C 20B3D720               FCC  / ｳﾗ /
0352  1F20 00                     FCB  $00
0353  1F21 52414D20       PD6:    FCC  /RAM /
0354  1F25 C6204153               FCC  /ﾆ AS/
0355  1F29 53454D42               FCC  /SEMB/
0356  1F2D 4C455220               FCC  /LER /
0357  1F31 CCDFDBB8               FCC  /ﾌﾟﾛｸ/
0358  1F35 DED7D120               FCC  /ﾞﾗﾑ /
0359  1F39 B6DE20B1               FCC  /ｶﾞ ｱ/
0360  1F3D D8CFBEDD               FCC  /ﾘﾏｾﾝ/
0361  1F41 0D0A00                 FCB  $0D,$0A,$00
0362  1F44 0D0A           PD7:    FCB  $0D,$0A
0363  1F46 BACFDDC4               FCC  /ｺﾏﾝﾄ/
0364  1F4A DE203F20               FCC  /ﾞ ? /
0365  1F4E 00                     FCB  $00
0366  1F4F 0A             PD8:    FCB  $0A
0367  1F50 31203A20               FCC  /1 : /
0368  1F54 CCDFDBB8               FCC  /ﾌﾟﾛｸ/
0369  1F58 DED7D120               FCC  /ﾞﾗﾑ /
0370  1F5C C3DDBFB3               FCC  /ﾃﾝｿｳ/
0371  1F60 0D0A                   FCB  $0D,$0A
0372  1F62 32203A20               FCC  /2 : /
0373  1F66 B5D3C320               FCC  /ｵﾓﾃ /
0374  1F6A 83E520B3               FCC  /▬▶ ｳ/
0375  1F6E D72020C3               FCC  /ﾗ  ﾃ/
0376  1F72 B7BDC420               FCC  /ｷｽﾄ /
0377  1F76 C3DDBFB3               FCC  /ﾃﾝｿｳ/
0378  1F7A 0D0A                   FCB  $0D,$0A
0379  1F7C 33203A20               FCC  /3 : /
0380  1F80 B3D72020               FCC  /ｳﾗ  /
0381  1F84 83E520B5               FCC  /▬▶ ｵ/
0382  1F88 D3C320C3               FCC  /ﾓﾃ ﾃ/
0383  1F8C B7BDC420               FCC  /ｷｽﾄ /
0384  1F90 C3DDBFB3               FCC  /ﾃﾝｿｳ/
0385  1F94 0D0A                   FCB  $0D,$0A
0386  1F96 41203A20               FCC  /A : /
0387  1F9A B1BEDDCC               FCC  /ｱｾﾝﾌ/
0388  1F9E DED720CD               FCC  /ﾞﾗ ﾍ/
0389  1FA2 20B3C2D9               FCC  / ｳﾂﾙ/
0390  1FA6 0D0A                   FCB  $0D,$0A
0391  1FA8 42203A20               FCC  /B : /
0392  1FAC 42415349               FCC  /BASI/
0393  1FB0 4320CD20               FCC  /C ﾍ /
0394  1FB4 D3C4DED9               FCC  /ﾓﾄﾞﾙ/
0395  1FB8 0D0A                   FCB  $0D,$0A
0396  1FBA 48203A20               FCC  /H : /
0397  1FBE BACFDDC4               FCC  /ｺﾏﾝﾄ/
0398  1FC2 DE20B2C1               FCC  /ﾞ ｲﾁ/
0399  1FC6 D7DD                   FCC  /ﾗﾝ/
0400  1FC8 0D0A00                 FCB  $0D,$0A,$00
0401  1FCB C3DDBFB3       PD9:    FCC  /ﾃﾝｿｳ/
0402  1FCF 20BCADB3               FCC  / ｼｭｳ/
0403  1FD3 D8AEB320               FCC  /ﾘｮｳ /
0404  1FD7 2121                   FCC  /!!/
0405  1FD9 0D0A00                 FCB  $0D,$0A,$00
0406  1FDC 54455854       PD10:   FCC  /TEXT/
0407  1FE0 20202648               FCC  /  &H/
0408  1FE4                PD11:   RMB  4
0409  1FE8 202D2026               FCC  / - &/
0410  1FEC 48                     FCC  /H/
0411  1FED                PD12:   RMB  4
0412  1FF1 0D0A00                 FCB  $0D,$0A,$00
0413  1FF4                END:    EQU  *
```

リスト２　ローダー

```
10 CLEAR 100,&H1C3F
20 LOADM"6809 A2":'     ｴﾃﾞｨﾀ･ｱｾﾝﾌﾞﾗ V2.0
30 LOADM"ALL RAM",,R:' ｺﾐﾆｹｰｼｮﾝ ﾌﾟﾛｸﾞﾗﾑ
40 END
```

FM-7 プログラム開発の手助けに！

F-BASIC DEBUGGING TOOL

デバッギング・ツール

■日原洋文

苦労の末やっと完成したプログラムをいざ実行してみると〝Illegal Function Call〟などのエラーが発生したり，プログラムは正常に（？）働くにもかかわらず，結果がどうもおかしいなどという経験をお持ちの方は多いと思います．こういうときは，まさに『シラミつぶし』いや，『南京虫つぶし』（？）でプログラムの流れを追いかけながら害虫駆除を行わなければなりません．

概　要

ここで発表するプログラムは，F-BASICに後述するコマンドを拡張するもので，次のような特徴があります．

❶ステートメント実行毎に指定されたサブルーチン（デバッグ・サブルーチン）にジャンプさせることができる．すなわち，F-BASICの**ON INTERVAL GOSUB**文が，一定時間間隔ごとの割り込みを行なうのと同様に，ステートメントごとの割り込みを行なえる．

❷デバッグ・サブルーチンへの分岐の許可，禁止を指定できる．

❸デバッグ・サブルーチンに分岐する直前の文のリストを，プログラムを中断することなく表示できる．

❹他のプログラムによりすでにF-BASICが拡張されている場合にも使用できる．

❺ROM-BASIC，Disk-BASICの区別なく動作する．

❻ポジション・インデペンデントである．

❼わずか0.5Kバイトである．

プログラムの入力，実行

上述したように本プログラムはリロケータブルですが，説明の都合上，リストどおりの位置（$6E00〜$6FFF）にあるものとします．

まず，**CLEAR,&H6DFF**として領域を確保した後，モニタを使ってリストどおり正しく入力します．

入力を終えたら，**SAVEM**″DEBUG″，**&H6E00**，**&H6FFF**，**&H6E00**としてセーブしておきます．

EXEC&H6E00で実行できます（次回からは，**CLEAR**文の後**LOADM** ″DEBUG″ **,,R**で可）．

コマンドの説明

●ON DEBUG GOSUB　行番号

機能：デバッグ・サブルーチンの定義をします．

解説：行番号はデバッグ・サブルーチンの開始行番号です．デバッグ・サブルーチンからの復帰は**DRET**文により行なわれます．

ON DEBUG GOSUB文の実行により，**DEBUG OFF**の状態になります．

●DEBUG ON/OFF

機能：デバッグ・サブルーチンへの分岐の許可，禁止を行ないます．

解説：**DEBUG ON**を実行すると，以後１ステートメント実行ごとに，**ON DEBUG GOSUB**文で指定されたデバッグ・サブルーチンを実行します．**DEBUG OFF**の実行後は分岐が行なわれません．

DEBUG ONの状態でダイレクト・モードのコマンドを入力するとエラーが発生することがあります．

●DRET

機能：デバッグ・サブルーチンから復帰します．

解説：デバッグ・サブルーチンからの復帰には必ず**DRET**文を用い，**RETURN**文は使いません（**RETURN**文を用いると，以後デバッグ・サブルーチンに分岐しなくなります）．

●DLIST

機能：デバッグ・サブルーチンに分岐する直前に実行した行のプログラム・リストを表示します．

解説：**LIST**文と異なり，**DLIST**文ではプログラムが中断されません．

DLIST文は，プログラム・リストの表示の前後において改行をおこなわないので，必要に応じ**PRINT**文を併用してください．

その他

デバッグ・サブルーチンに分岐する直前に実行した文の行番号は，プログラムの最初で

DEF FNDB＝PEEK(1)＊256＋PEEK(2)

としておけば，以後はFNDBで得ることができます．ただ

し，ここでの行番号は，実行したばかりの文の行番号であって，F-BASICの**TRON**文のように，これから実行しようとする文の行番号ではないので注意してください．

以上の拡張コマンドを上手に使うことにより，BASICのステップ・トレース，特定の変数の値の追跡などが可能となり，デバッグ作業が極めて容易となることでしょう．

◘参考文献

1) 世田谷マイコンクラブ：〝FM-8システムエリアの解析〟，I/O'81年12月号

2) 平井秀明：〝BASIC ANALIZER〟，FM-7/8活用研究

リスト1　FM-7用プログラム・ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6E00 12 C6 39 E7 8C FA C6 03 B6 01 EB 2B 01 5A 86 80 :75
6E10 CE 01 E5 33 4A AB C4 5A 26 F9 81 FB 23 04 C6 15 :97
6E20 20 54 A7 8C 71 AE 43 AF 8C 6D 30 8C 6C AF 43 17 :E2
6E30 00 FD AE 41 31 8D 00 F7 10 AF 41 A6 C4 8D 1A 30 :E2
6E40 8C 23 86 03 8D 13 A6 C4 8B 03 A7 C4 BE 01 ED AF :96
6E50 8C 42 30 8C 56 BF 01 ED 39 26 01 39 E6 80 E7 A0 :13
6E60 2A FA 4A 20 F4 44 45 42 55 C7 44 52 45 D4 44 4C :A8
6E70 49 53 D4 7E 8D AA 7E 8D D1 7E 8F 1C 7E 90 71 7E :27
6E80 9B DB 7E 9C 22 7E B6 15 7E C1 69 7E C6 3D 7E D4 :76
6E90 11 7E D9 0F 00 00 00 00 00 9D D8 A0 8C F8 27 4A :81
6EA0 4A 27 68 4A 27 6C 9D D8 6E 9C EC A1 8C E8 27 03 :60
6EB0 6E 9C E1 30 8C 11 BF 02 64 0F 00 C6 39 F7 02 63 :47
6EC0 9D D2 8E 00 00 20 C7 0D 00 26 1E C6 04 8D A4 DC :0C
6ED0 47 DD 01 34 06 DC D9 34 06 4F 5F 34 06 86 CE 34 :BE
6EE0 02 DC 03 DD D9 03 00 20 A2 39 9D D2 26 04 C6 16 :0A
6EF0 20 84 81 97 27 09 81 D5 27 09 C6 02 16 FF 77 C6 :8C
-----------------------------------------------------------
Sum: F5 F5 FA E1 B7 A3 6A A8 81 44 65 16 18 A9 AF 65 :46
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6F00 7E 20 02 C6 39 F7 02 63 9D D2 39 0F 00 9D D2 16 :37
6F10 FF 6A 9E 01 9F 4B 17 FF 60 9F A2 17 FF 67 17 FF :3C
6F20 61 9E A2 17 FF 62 8E 04 3D 17 FF 65 9D D2 39 34 :3F
6F30 40 30 8C 04 17 FF 48 35 C0 0D 0A 11 03 20 20 20 :DE
6F40 20 20 2D 2D 2D 20 46 2D 42 41 53 49 43 20 44 45 :65
6F50 42 55 47 47 49 4E 47 20 54 4F 4F 4C 20 2D 2D 2D :08
6F60 0D 0A 0D 0A 11 04 20 20 53 54 41 54 45 4D 45 4E :E4
6F70 54 3A 20 4F 4E 20 44 45 42 55 47 20 47 4F 53 55 :30
6F80 42 20 3C 6C 69 6E 65 20 6E 6F 2E 3E 0D 0A 20 20 :06
6F90 20 20 20 20 20 20 20 20 20 20 20 44 45 42 55 47 :C7
6FA0 20 4F 4E 2F 4F 46 46 0D 0A 20 20 20 20 20 20 20 :BE
6FB0 20 20 20 20 20 20 44 52 45 54 0D 0A 20 20 20 20 :86
6FC0 20 20 20 20 20 20 20 20 20 44 4C 49 53 54 0D 0A :B7
6FD0 11 06 20 20 45 78 65 63 2E 20 4C 4E 2E 3A 20 44 :90
6FE0 45 46 20 46 4E 44 42 3D 50 45 45 4B 28 31 29 2A :D3
6FF0 32 35 36 2B 50 45 45 4B 28 32 29 0D 0A 11 07 00 :9F
-----------------------------------------------------------
Sum: 2B 61 CF 3B BE 4A FB F7 C8 AC 8F 40 D3 3B 5D 9D :DB
```

FM-7

マイクロvi

micro-vi

ASCIIファイルのフルスクリーン・エディタ

■津田伸秀

micro-viは，バークレイ版UNIXのviに準拠した，FM-7のF-BASIC用のフルスクリーン・エディタです．

UNIXviは，16種，140以上ものコマンドを持ち，非常に強力で，キーストロークが少なく，使い勝手が良く，かつものすごく大きなエディタです．このように大きなものをそのまま6809上で実現するには，64Kというメモリ空間はあまりにも狭すぎます．それで作者が特によく使うコマンドにしぼって仕様を決め，作成したのがmicro-viです．

micro-viはほぼUNIXのサブセットになっていますが，作者の好みに合わないところや，単に面倒でプログラムを作らなかったところなど，一部の仕様がviと異なっているので，UNIXviに習熟している人が，viのつもりで使うと戸惑うことがあるかも知れません．そのような人は，マニュアルをよく読んで，viとの違いをよく頭に入れてから使ってください．

そうは言っても，micro-viは60ものコマンドを持ち，充分使い勝手の良い実用的なフルスクリーン・エディタです．

エディタの起動と終了

RUN"vi" ⏎ によりviが起動します．

File Name :

エディットをするファイルの名前を聞いてくるので，ファイル名を入力するとviのエディット・モードになります．

viに入り，エディットを行なった後(もしくはまったく何も行なわず)，BASICに戻るにはコマンド・モードにおいて Z Z とキーインします．この操作によりファイルは更新され，BASICのコマンド・モードに戻ります．

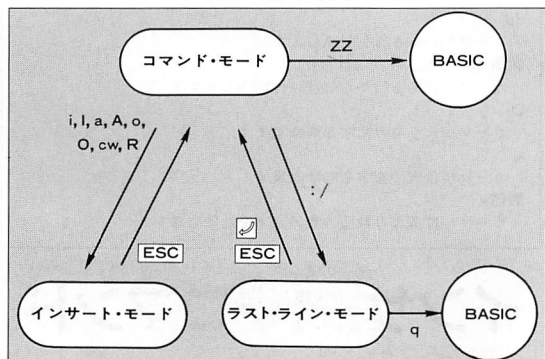



図1　表示形式

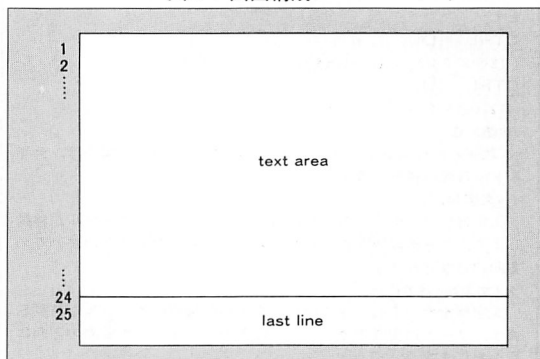
FLEXeditorでは，エディタを起動するごとにバックアップ・ファイルを作ってくれますが，UNIXと同じようにある面で不親切なviは，そのようなことはやってくれません．したがって重要なファイルのバックアップは，自分で行なってください．また，viでのファイルの更新は，Z Z をキーインした時点で行なわれるので，エディット中にリセットなどでviから抜け出した場合，ディスク中のファイルはviに入る直前のもののまま，要するに何も行なわれなかったことになります．

画面構成

viの画面は図2のように，テキスト・エリアとラスト・ラインと呼ばれる部分の2つに分かれています．

画面の1行目から24行目までの領域は，テキスト・エリアと呼ばれます．このエリアには編集するテキストが表示されます．テキスト・エリアの1箇所は反転文字になっており，これをカーソルと呼びます．カーソルはカーソル移動コマンドにより自由に移動させることができます．カーソルが画面からはみ出した場合は，スクロールまたは逆スクロールを行ないます．

ラスト・ラインは，viからのメッセージをオペレータに伝えたり，サーチ・コマンドのように文字列をパラメータとして持つコマンドや，特殊なコマンドのときに使う行です．

図2　画面構成

viのモード

viには3つのモードがあります．これらは明確に区別され，コマンドにより自由にモードを変えることができます．

❶Command mode（コマンド・モード）

通常の状態で，viの起動時はこのコマンド・モードになります．viに使用されているいろいろなコマンドが使えます．

❷Insert mode（インサート・モード）

a, A, i, I, o, O, cw, Rによりインサート・モードになります．上記コマンドに続けてキーインした文字は[↵]も含め，すべてテキストにインサートされます．インサート・モードから抜けるには[ESC]を使います．

❸Last line mode（ラスト・ライン・モード）

[:]または[/]によりカーソルが最下行に移動します．文字列サーチなどのように，文字列をパラメータとするコマンドや特殊なコマンドのとき，このモードとなります．

viのコマンド

micro-viのコマンドは，カーソル移動コマンド34個，デリートのコマンド4個，インサート・コマンド7個，リプレイス・コマンド2個，アジャスト・コマンド3個，その他の特殊なコマンド11個と，とても全部を一度に覚えきれないくらいのたくさんのコマンドを用意しています．

カーソル移動コマンド

エディタでテキストを修正するときは，修正したい場所までカーソルを移動させ，**cw**などのコマンドを使って，悪いところを修正します．また，テキストの確認をするときもカーソルを移動させることで，ウィンドウを移動させて確認をします．つまり，エディタを使っているときのかなりの時間は，カーソルの移動に費やされていることになります．したがって，目的とする場所にいかに早くカーソルを移動できるかが，エディタの性能を評価するにおいてかなり大きな要因となります．

micro-viでは，カーソル移動のために34ものコマンドを用意して，目的とする場所へす早くカーソルを移動させることができます．

カーソル移動コマンド

[CTRL]+[F]
windowを1ページ進めます．すなわち24行スクロールします．

[CTRL]+[B]
windowを1ページ前に戻します．すなわち24行逆スクロールします．

[CTRL]+[D]
12行スクロールし，windowが半ページ進みます．

[CTRL]+[U]
12行逆スクロールし，windowが半ページ前に戻ります．

〈数値〉G
〈数値〉の行にジャンプします．〈数値〉を省略した場合は，テキストの最後に移動します．

／〈文字列〉[↵]
文字列をサーチし，みつかればパターンの場所にカーソルが移動します．サーチは現在のカーソルの位置から，テキストの終わりの方向に行なわれます．

／〈文字列〉[ESC]
文字列をサーチし，みつかれば文字列の場所にカーソルが移動します．サーチは現在のカーソルの位置からテキストの始めの方向に行なわれます．

n
サーチ・コマンドで入力した文字列と同じものをサーチします．サーチは現在のカーソル位置よりテキストの終わりの方向に行なわれます．

N
サーチ・コマンドで入力したパターンと同じものをサーチします．サーチは現在のカーソルの位置からテキストの始まりの方向に行なわれます．

Hまたは[HOME]
カーソルをウィンドウの先頭に移動させます．

M
カーソルをウィンドウの中央に移動させます．

L
カーソルをウィンドウの最後に移動させます．

＋または[↵]
カーソルを次の行に移動させます．文の先頭のスペースはスキップします．

－
カーソルを直前の行に移動させます．文の先頭のスペースはスキップします．

0
現在カーソルがある行の，最初のスペースでない文字のところにカーソルが移動します．

$
現在カーソルがある行の最後尾にカーソルが移動します．

jまたは[↓]
次の行の同じカラムにカーソルが移動します．

kまたは[↑]
直前の行の同じカラムにカーソルが移動します．

lまたは[→]または[SPACE]
カーソルが1つ右に進みます．

hまたは[←]または[BS]
カーソルが1つ左に戻ります．

w
次の語の先頭にカーソルが移動します．

b
直前の語の先頭にカーソルが戻ります．

e
次の語の最後にカーソルが移動します．

W
次の，スペースで区切られる文字列の先頭にカーソルが移動します．

B
直前の，スペースで区切られる文字列の先頭にカーソルが移動します．

E
次の，スペースで区切られる文字列の最後にカーソルが移動します．

デリート・コマンド

不必要な文字列を文字単位，行単位で削除できます．

dw
カーソルの位置の語を削除します．

dd
カーソルのある行を削除します．

数値dd
カーソルのある行から〈数値〉行削除します．

D
カーソルより右の文字を削除します．

x
カーソル位置の文字を削除します．

数値x
カーソル位置より数値の数の文字を削除します．

インサート・コマンド

ノーマル・モードから，**i, I, a, A, o, O, cw**によりインサート・モードになります．インサート・モードに入るにはこのように7種の方法があり，場合により使いわけてください．インサート・モードから抜けるには ESC をキーインします

i
　カーソルのすぐ左に文字を挿入します．iによりインサート・モードに入ると，カーソルは移動せずに入力待ちとなります．文字をキーインすんと，カーソルの位置に文字が挿入され，カーソルが1つ右に進みます．インサート・モード中はカーソル・キーは使えません．間違ってキーを押した場合は BS で1文字抹消できます．⏎を押すと新しい行が挿入され，カーソルは直前の行と同じカラムに移動します．
　インサート・モードから抜け，ノーマル・モードに戻るには，ESC をキーインします．
I
カーソルのある行の先頭に挿入します．
Iをキーインすると，カーソルがその行の最初の空白でない文字のところに移動し，インサート・モードとなります．その後のオペレーションは，iコマンドと同様です．
a
カーソルのすぐ右に挿入します．
aをキーインすると，カーソルが右に1つ移動してインサート・モードになります．その後のオペレーションはiコマンドと同様です．
A
カーソルのある行にアペンドします．
Aをキーインすると，カーソルが行末に移動してインサート・モードになります．その後のオペレーションはiコマンドと同様です．
o
カーソルのある行の次に挿入します．
oをキーインすると，カーソルの下の行が1行ずつスクロールし，すぐ下に空文ができます．カーソルの下の行の先頭と同じカラムに移動し，インサート・モードになります．その後のオペレーションはiコマンドと同様です．
O
カーソルのある行の直前に挿入します．
小文字のoコマンドは，次の行に空文を作りますが，大文字のOコマンドは，前の行に空文を作ります．それ以外はoコマンドと同様です．
cw
カーソル位置の語を消去し，インサート・モードになります．1語を書き変えたいときにたいへん便利です．

リプレイス・コマンド

　インサート・コマンドは文字をキーインした分だけ元のテキストが右にずれ，カーソルの位置の文字は消去されませんが，リプレイス・コマンドではFLEXeditorのオーバレイ・コマンドのようにカーソル位置の文字を書き変えます．このコマンドはある意味で危険なコマンドなので，注意して使ってください．

r
カーソル位置の文字を1文字書き変えます．
rに続けて入力した文字が，カーソル位置の文字にとって変わります．
R
カーソル位置以後の文字を連続的に書き変えます．
Rをキーインした後，文字をキーインするごとにカーソル位置の文字が置き変わり，カーソルが1つ右に移動します．リプレイス・モードから抜け出すには，インサート・モード同様に ESC をキーインします．

アジャスト・コマンド

　2行に渡る行を1つにしたいとき，多くの行の字下げを変化させるコマンドを用意しています．

J
　カーソルのある行にすぐ次の行を結合します．
新しくできる行が80文字を越える場合は，79文字以降が消去されます．
>>
　カーソルのある行を，タブの数だけ右にシフトします(micro-viの場合，タブのデフォルトは4)．>>の直前に数字をキーインしたときは，その数だけ右にシフトします．
<<
>> コマンドと同様の働きをします．ご想像どおり右でなく左にシフトを行ないます．

コロン・コマンド

　コロンをキーインすると，カーソルが24行目のラスト・ラインに移動し，1行入力モードとなります．このモードではブロック転送のように，いくつかのパラメータを必要とする特殊で便利なコマンドです．
　コロン・コマンドでは，コマンドに先だって行番号を入力することができます．行番号は数字そのものに加え，〝.〞(ピリオド)で現在カーソルがある行，〝$〞で最後の行を表します．さらに演算子〝+〞，〝−〞が使用可能です．たとえば，

＋2

は現在カーソルがある行の2行後の行を表します．

：<line>⏎
　lineにカーソルが移動します．
例

：1	1行目にジャンプします．
：$	テキストの最後の行にジャンプします．
：$−20	最後から20行目にジャンプします．
：．+100	現在カーソルがある行から100行目にジャンプします．

：<line1>, <line2>m<line3>⏎
　line1からline2までの行をline3の直後に移動します．
　(line1<line2) and (line3<line1 or line3>line2) でなくてはいけません．またline2を省略した場合は，line2＝line1とみなします．
例

：10, 20m40	10行目から20行目までが40行目の次に移動します．
：.m. +10	現在カーソルのある行がその行の10行後に移動します．
：1, . m$	1行目から現在カーソルがある行までがテキストの最後に移動します．つまり現在行の前後がそっくり入れ替わります．

：<line1>, <line2>co<line3>⏎
　line1からline2までをline3の直後にコピーします．(line1<line2) and (line3<line1 or line3>line2) でなくてはいけません．また，line2を省略した場合は，line2＝line1とみなします．
例

：10, 20co30	10行目から20行目までを30行目のつぎにコピーします．
：., . +10co. +10	現在カーソルがある行から11行を10行後にコピーします．すなわち11行の複製が作られます．

：<line1>, <line2>s/pat1/pat2⏎
　line1からline2までの間で，最初のpat1をpat2に変換します．文字列のセパレータは通常 / を使いますが，これはpat1，pat2に含まれない文字ならば何でも構いません．
例

：10, 20s/kumagai/ikejiri/	10行目から20行までの間の最初のkumagaiをikejiriに変えます．
：1, $s/go to/makichan/	テキストの最初のgo toをmakichanに

	変えます.

:<line1>, <line2>s/pat1/pat2/g⏎
line1からline2までの間のすべてのpat1をpat2に変換します.
例

:1, $s /Z80/6809/g	すべてのZ80を6809に変えます.
:100, 200s/long skirt/mini skirt/g	100行目から200行までのすべてのlong skirtをmini skirtに変換します.

:<line1>, <line2>d⏎
line1からline2までの行を消去します.
:q⏎
テキストをセーブせずに, viから抜け出します.
:<数値>t⏎
水平タブのカラム数を設定します.

その他のコマンド

?
現在の行番号と, 残りのメモリサイズを表示します.

UNIXviを使っている人への注意

既に述べたように, micro-viはUNIXviに準拠していますが, 完全にコンパチブルではありません. micro-viはできる限りUNIXviとのコンパチビリティを確保するようつとめましたが, 一部の仕様がviとは異なっています.

以下にmicro-viの相違点, 注意点を示します.

❶**undo**コマンドがありません.

micro-viではかなりの数のコマンドが省略されています. 特に, **u**コマンドがないのでUNIXviのように, 間違って消してしまったラインを復活させることができません.

注意してください.

❷サーチコマンドの仕様が少し異なります.

テキストの始まりの方に向かってサーチするときの形式が異なっています(UNIXviは?pat⏎だがmicro-viは/pat **ESC**). またUNIXviはサーチのときにテキストの最後まで行くと, テキストの始めにワープしますが, micro-viではワープしません.

❸パターン指定のとき特殊キャラクタが使えません.

サーチ・コマンドなどのパターンで"*", ",", "."などの特殊文字は使えません.

❹インサート・モードの中に **CTRL**+**D** は使えません. その代わりに **BS** が正常に動作します.

❺左または右へのカーソル移動で行を変ることができます.

UNIXviでは, 行末でさらに右にカーソルを移動させることはできませんが, micro-viでは次の行に移ることが可能です.

❻UNIXviでは, 水平タブは水平タブのコードでメモリに収納されますが, micro-viではすべてスペースに変わってメモリに格納されます.

表1 マシン語プログラム・エリア

プログラム名	開始	終了	実行	リスト	備考
vim	2000	53FF	2000	2	マシン語メイン
vil. O	1400	1906	1700	3	ローダ(コメント付)
vis. O	1400	1879	1700	4	セーバ(コメント付)
vil2. O	1400	186A	1700	5	ローダ(コメントなし)
vis2. O	1400	1859	1700	6	セーバ(コメントなし)

注意

❶80字を越える行を含むファイルは編集できません.

❷ディスク・ドライブは, ドライブ0のみを使ってください.

❸ASCII形式のファイルを扱います.

BASICのコメント付ファイルの場合は, ローダとして**リスト3**の"vil. O"を使い, セーバとして**リスト4**の "vis. O"を使ってください. このときのBASICメイン・プログラムを**リスト1**に示します. アセンブラ(富士通製)やKコンパイラなどのソース・プログラムを作るのに利用できます.

単なるASCIIファイルの場合は, ローダとして**リスト5**の "vil2. O" を使い, セーバとして**リスト6**の"vis2. O"を使ってください. このときのBASICメイン・プログラムは, **リスト1**の190行と210行を変更してください(**リスト7**).

viを起動するには, **リスト1**のBASICメイン・プログラムを, **RUN"vi"**⏎で走らせてください.

"File Name :" と聞いてきたら, 編集したいファイル名を入力してください. 編集を終らせると, "vi. out" というファイルが作られ, 次に指定のファイル名に名前が変更されます.

最後に

FLEX用のmicro-viが, 制作ワンダーソフト, 販売コムパックから, ディスケットで供給されています. 価格は¥9,800です. こちらはFLEX用ということで, すべてマシン語です.

リスト1 vi BASICプログラム

```
100 '  ┌─────────────────────┐
110 '  | Micro-vi  Version 1.0 |
120 '  | Copyright (c) 1983 by |
130 '  | N.Tsuda               |
140 '  └─────────────────────┘
150 '
160 CLEAR 50,&H13FF:ON ERROR GOTO 240:CONSOLE,,,1
165 POKE &H6FFB,&H3B:POKE &HFFF6,&H6F:POKE &HFFF7,&HFB
170 WIDTH80,25:POKE &H6FFC,&H80:POKE &H6FFD,0
180 INPUT"File Name :",F$:IF F$="" GOTO 180
190 OPEN"I",#1,F$:LOADM"vil2.O",,R:CLOSE #1
200 LOADM"vim",,R:IF PEEK(&H6FFC)=0 THEN 230
210 OPEN"O",#2,"vi.out":LOADM"vis2.O",,R:CLOSE #2
220 KILL F$:NAME "vi.out"AS F$
230 CLEAR 300,&H6FFF:CONSOLE,,,0:POKE &HFFF6,1:POKE &HFFF7,&HE0:END
240 IF ERL=190 THEN CLOSE #1:LOCATE 0,24:PRINT F$": New File";:RESUME 200
250 IF ERL=220 THEN RESUME NEXT
```

リスト2 vimマシン語ダンプ・リスト

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2000 34 7F 10 FF 6F FE 8E 6F 00 32 01 17 01 6C 32 E9 :FE
2010 FF 00 33 E9 00 80 31 8D 33 BB 17 01 9D 17 01 A3 :B7
2020 10 FE 6F FE 35 FF 34 70 17 01 1A 1F 89 C1 1A 26 :2E
2030 05 CC FF FF 35 F0 C1 0D 26 02 C6 0A 4F 35 F0 EC :1A
2040 62 34 76 1F 98 81 09 26 02 86 20 81 0A 26 05 17 :E8
2050 01 28 35 F6 17 01 29 35 F6 EC 62 34 06 8D E0 35 :EA
2060 86 34 36 A6 E4 E6 63 3D E7 64 EC 61 3D E7 65 A6 :C7
2070 61 E6 63 3D AB 64 AB 65 32 66 39 6F E2 8C 00 00 :B4
2080 2A 09 63 E4 1E 01 17 00 95 1E 01 4D 2A 05 63 E4 :27
2090 17 00 8B 17 00 8D 1F 10 6D E0 2A 03 17 00 7F 39 :BE
20A0 17 00 80 1F 10 39 6F E2 8C 00 00 2A 06 1E 01 8D :B8
20B0 6D 1E 01 4D 2A 04 63 E4 8D 64 8D 67 6D E0 2A 02 :AC
20C0 8D 5C 39 8D 5E 39 8C 00 00 2B 08 27 0E 58 49 30 :0B
20D0 1F 20 F8 27 06 47 56 30 01 20 F8 39 8C 00 00 2B :3A
20E0 08 27 0E 58 49 30 1F 20 F8 27 06 44 56 30 01 20 :5D
20F0 F8 39 8C 00 00 2B 08 27 0E 47 56 30 1F 20 F8 27 :50
-----------------------------------------------------------
Sum: 03 C2 2F 50 1C DF 05 C3 A3 47 B3 7B 68 4A D6 DE :85

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2100 06 58 49 30 01 20 F8 39 8C 00 00 2B 08 27 0E 44 :61
2110 56 30 1F 20 F8 27 06 58 49 30 01 20 F8 39 40 50 :9D
2120 82 00 39 34 16 4F 5F 8E 00 11 A3 62 24 02 E3 62 :C2
2130 69 61 69 E4 59 49 30 1F 26 F0 46 56 63 61 63 E4 :C5
2140 35 10 32 62 39 8D 50 CC 0C 03 FD FC 82 8D 58 8D :B7
2150 46 CC 29 00 FD FC 82 8D 4E 8D 3C 86 80 B7 FC 80 :93
2160 FC FC 83 34 06 8D 40 35 06 5D 27 D9 34 02 8D 27 :04
2170 CC 0C 02 FD FC 82 8D 2F 35 82 86 0D 8D 02 86 0A :7A
2180 34 12 8D 13 8E FC 80 A6 E4 A7 04 86 03 A7 02 86 :DD
2190 01 A7 03 8D 12 35 92 B6 FD 05 2B FB 86 80 B7 FD :A9
21A0 05 B6 FD 05 2A FB 39 4F B7 FD 05 39 1A 40 39 1C :0B
21B0 BF 39 B6 FD 0F 39 B7 FD 0F 39 CC 00 00 ED A8 21 :71
21C0 16 32 11 34 40 33 E4 32 7E 4F 5F ED A4 CC 00 01 :A0
21D0 ED 22 CC 00 02 ED 24 CC 00 03 ED 5E EC 5E 83 00 :D5
21E0 10 10 2C 00 17 EC 5E 58 49 30 A4 30 8B CC FF FF :A7
21F0 ED 84 EC 5E C3 00 01 ED 5E 16 FF E0 EC 46 34 06 :2B
-----------------------------------------------------------
Sum: 83 5D 22 2F 95 E8 95 E6 5C 1A BF 80 F4 9B 4B DE :96

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2200 EC 44 34 06 17 06 DF 32 64 35 C6 34 40 33 E4 EC :6E
2210 46 83 00 00 10 26 00 0E CC 00 01 ED A8 21 E6 45 :BB
2220 1D E7 A8 20 35 C0 CC 00 01 AE 46 E7 88 3E E6 45 :5A
2230 1D AE 46 E7 88 3D 35 C0 34 40 33 E4 EC 44 10 83 :00
2240 00 00 10 26 00 17 EC A8 21 10 27 00 0B 4F 5F ED :DF
2250 A8 21 E6 A8 20 1D 35 C0 17 FD CB 35 C0 10 83 00 :F0
2260 02 27 0E 10 83 00 01 27 08 10 83 FF FF 10 26 00 :C1
2270 05 CC FF FF 35 C0 35 C0 34 40 33 E4 EC 46 10 83 :09
2280 00 00 27 08 10 83 FF FF 10 26 00 05 CC FF FF 35 :FA
2290 C0 10 83 00 01 10 26 00 0C E6 45 1D 34 06 17 FD :2C
22A0 9E 32 62 35 C0 10 83 00 02 10 26 00 0C E6 45 1D :46
22B0 34 06 17 FD A4 32 62 35 C0 35 C0 34 40 33 E4 EC :E7
22C0 A4 34 06 E6 45 1D 34 06 17 FF 40 32 64 35 C0 34 :75
22D0 40 33 E4 EC A4 34 06 17 FF 5E 32 62 35 C0 34 40 :92
22E0 33 E4 EC 22 34 06 E6 45 1D 34 06 17 FF 8A 32 64 :17
22F0 35 C0 34 40 33 E4 30 46 34 10 EC 44 34 06 EC 22 :B2
-----------------------------------------------------------
Sum: F9 C3 52 58 81 2D 91 2B 1E 72 77 49 2A 2E 29 9E :3F

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2300 34 06 17 00 1B 32 66 35 C0 34 40 33 E4 30 48 34 :30
2310 10 EC 46 34 06 EC 44 34 06 17 00 04 32 66 35 C0 :8E
2320 34 40 33 E4 32 71 AE 46 E6 80 AF 46 1D E7 59 83 :5D
2330 00 00 10 27 01 A2 E6 59 1D 83 00 25 10 27 00 11 :26
2340 EC 44 34 06 E6 59 1D 34 06 17 FF 2C 32 64 16 01 :EF
2350 84 E6 D8 06 1D 83 00 2D 10 27 00 04 4F 5F 20 03 :21
2360 CC 00 01 ED 5E 83 00 00 10 27 00 07 EC 46 C3 00 :CE
2370 01 ED 46 E6 D8 06 1D 34 06 17 03 A0 32 62 83 00 :20
2380 00 10 27 00 18 EC 46 34 06 30 5C 34 10 17 01 4C :EF
2390 32 64 34 06 EC 46 E3 E1 ED 46 16 00 04 4F 5F ED :AE
23A0 5C E6 D8 06 1D 83 00 2E 10 26 00 07 EC 46 C3 00 :20
23B0 01 ED 46 E6 D8 06 1D 34 06 17 03 60 32 62 83 00 :E0
23C0 00 10 27 00 18 EC 46 34 06 30 5A 34 10 17 01 0C :AD
23D0 32 64 34 06 EC 46 E3 E1 ED 46 16 00 05 CC 7F FF :5E
23E0 ED 5A AE 46 E6 80 AF 46 1D E7 59 10 83 00 64 10 :FA
23F0 26 00 14 30 51 34 10 AE 48 EC 81 AF 48 34 06 17 :AA
-----------------------------------------------------------
Sum: 89 5E 89 8C C1 37 A6 1D 56 C6 B0 07 F4 34 E2 F7 :8B

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2400 02 00 32 64 16 00 B5 10 83 00 6F 10 26 00 14 30 :DF
2410 51 34 10 AE 48 EC 81 AF 48 34 06 17 02 4C 32 64 :24
2420 16 00 99 10 83 00 78 10 26 00 14 30 51 34 10 AE :77
2430 48 EC 81 AF 48 34 06 17 02 48 32 64 16 00 7D 10 :80
2440 83 00 75 10 26 00 14 30 51 34 10 AE 48 EC 81 AF :19
2450 48 34 06 17 01 E7 32 64 16 00 61 10 83 00 63 10 :94
2460 26 00 14 30 51 34 10 AE 48 EC 81 AF 48 34 06 17 :AA
2470 01 E3 32 64 16 00 45 10 83 00 73 10 26 00 20 EC :1D
2480 5A 34 06 EC 5C 34 06 EC 5E 34 06 AE 48 EC 81 AF :AC
2490 48 34 06 EC 44 34 06 17 00 8D 32 6A 16 FE 87 10 :D7
24A0 83 00 00 10 26 00 04 32 C4 35 C0 30 51 34 10 E6 :53
24B0 59 1D 34 06 17 01 9E 32 64 16 00 00 EC 5A 34 06 :92
24C0 EC 5C 34 06 EC 5E 34 06 30 51 34 10 EC 44 34 06 :35
24D0 17 00 54 32 6A 16 FE 4E 32 C4 35 C0 34 40 33 E4 :DF
24E0 32 7E 4F 5F ED 5E ED D8 04 E6 D8 06 1D 34 06 17 :A4
24F0 02 2A 32 62 83 00 00 10 27 00 28 AE 46 E6 80 AF :AB
-----------------------------------------------------------
Sum: 58 C0 66 73 5A 76 1C DB 38 A3 81 04 E6 B6 16 6F :39
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2500 46 1D 34 06 EC D8 04 58 49 34 06 58 49 58 49 E3 :65
2510 E1 E3 E1 83 00 30 ED D8 04 EC 5E C3 00 01 ED 5E :7A
2520 16 FF C6 EC 5E 35 D0 34 40 33 E4 32 7E EC 46 34 :CB
2530 06 17 03 8F 32 62 A3 4C 10 2C 00 0C EC 46 34 06 :E6
2540 17 03 80 32 62 16 00 02 EC 4C ED 5E EC 5E A3 4A :00
2550 10 2C 00 07 EC 4A A3 5E 16 00 02 4F 5F ED 4A EC :63
2560 48 10 27 00 21 EC 4C 34 06 EC 46 34 06 EC 44 34 :E2
2570 06 17 00 32 32 66 EC 4A 34 06 EC 44 34 06 17 00 :D8
2580 5C 32 64 16 00 1E EC 4A 34 06 EC 44 34 06 17 00 :17
2590 4C 32 64 EC 4C 34 06 EC 46 34 06 EC 44 34 06 17 :41
25A0 00 04 32 66 35 C6 34 40 33 E4 E6 D8 06 1D 10 27 :3A
25B0 00 29 EC 48 C3 FF FF ED 48 83 00 00 10 2D 00 15 :28
25C0 EC 44 34 06 AE 46 E6 80 AF 46 1D 34 06 17 FC A8 :CB
25D0 32 64 16 00 03 16 00 03 16 FF CF 35 C0 34 40 33 :48
25E0 E4 EC 46 C3 FF FF ED 46 83 00 00 10 2D 00 11 EC :C7
25F0 44 34 06 CC 00 20 34 06 17 FC 7D 32 64 16 FF E1 :C0
-----------------------------------------------------------
Sum: A6 C5 01 B4 11 E3 6B C0 2D 9F AA 31 1D AD 71 E0 :01

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2600 35 C0 34 40 33 E4 EC 44 83 00 00 10 2C 00 1F CC :5A
2610 00 2D AE 46 E7 80 AF 46 EC 46 34 06 EC 44 40 50 :A9
2620 82 00 34 06 17 00 16 32 64 C3 00 01 35 C0 EC 46 :6A
2630 34 06 EC 44 34 06 17 00 04 32 64 35 C0 34 40 33 :F1
2640 E4 CC 00 0A 34 06 EC 46 34 06 EC 44 34 06 17 00 :E1
2650 63 32 66 35 C0 34 40 33 E4 E6 45 1D AE 46 E7 84 :22
2660 4F 5F AE 46 E7 01 EC 46 35 C0 34 40 33 E4 CC 00 :08
2670 08 34 06 EC 46 34 06 EC 44 34 06 17 00 36 32 66 :FD
2680 35 C0 34 40 33 E4 CC 00 10 34 06 EC 46 34 06 EC :EE
2690 44 34 06 17 00 1E 32 66 35 C0 34 40 33 E4 EC 44 :FB
26A0 83 00 0A 10 2C 00 06 CC 00 30 16 00 03 CC 00 37 :E7
26B0 E3 44 35 C0 34 40 33 E4 32 74 30 56 1F 10 ED 54 :43
26C0 4F 5F AE 54 E7 80 AF 54 EC 48 34 06 EC 44 35 10 :FD
26D0 17 F9 F0 34 06 17 FF C2 32 62 AE 54 E7 80 AF 54 :12
26E0 EC 48 34 06 EC 44 35 10 17 F9 B5 ED 44 83 00 00 :5C
26F0 10 26 FF D4 4F 5F ED 5E AE 54 E6 82 AF 54 1D AE :3A
-----------------------------------------------------------
Sum: CA 82 66 CA 41 55 ED 01 C2 AA 00 4F 83 2D 67 4C :1E

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2700 46 E7 80 AF 46 83 00 00 10 27 00 0A EC 5E C3 00 :73
2710 01 ED 5E 16 FF E2 EC 5E 32 C4 35 C0 34 40 33 E4 :03
2720 E6 45 1D 34 06 CC 00 30 A3 E1 10 2E 00 0A E6 45 :75
2730 1D 83 00 39 10 2F 00 04 4F 5F 20 03 CC 00 01 35 :EF
2740 C0 34 40 33 E4 E6 45 1D 83 00 20 10 27 00 18 E6 :6B
2750 45 1D 83 00 09 10 27 00 0E E6 45 1D 83 00 0A 10 :18
2760 27 00 04 4F 5F 20 03 CC 00 01 35 C0 34 40 33 E4 :49
2770 E6 45 1D 34 06 17 00 2D 32 62 83 00 00 10 26 00 :13
2780 1F E6 45 1D 34 06 17 00 41 32 62 83 00 00 10 26 :46
2790 00 0E E6 45 1D 83 00 5F 10 27 00 04 4F 5F 20 03 :44
27A0 CC 00 01 35 C0 34 40 33 E4 E6 45 1D 34 06 CC 00 :9B
27B0 41 A3 E1 10 2E 00 0A E6 45 1D 83 00 5A 10 2F 00 :71
27C0 04 4F 5F 20 03 CC 00 01 35 C0 34 40 33 E4 E6 45 :4D
27D0 1D 34 06 CC 00 61 A3 E1 10 2E 00 0A E6 45 1D 83 :1B
27E0 00 7A 10 2F 00 04 4F 5F 20 03 CC 00 01 35 C0 34 :84
27F0 40 33 E4 E6 45 1D 34 06 17 FF CF 32 62 83 00 00 :D5
-----------------------------------------------------------
Sum: E9 F9 45 90 34 98 E2 67 ED C0 7B 08 23 4E 46 5D :10

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2800 10 27 00 09 E6 45 1D C3 FF E0 16 00 03 E6 45 1D :8B
2810 35 C0 34 40 33 E4 E6 45 1D 34 06 17 FF 87 32 62 :33
2820 83 00 00 10 27 00 09 E6 45 1D C3 00 20 16 00 03 :07
2830 E6 45 1D 35 C0 34 40 33 E4 32 7C E6 D8 04 1D 34 :89
2840 06 17 FE FD 32 62 83 00 00 10 27 00 0A EC 44 C3 :63
2850 00 01 ED 44 16 FF E4 E6 D8 04 1D 83 00 2D 10 27 :F1
2860 00 04 4F 5F 20 03 CC 00 01 ED 5C 83 00 00 10 27 :A5
2870 00 07 EC 44 C3 00 01 ED 44 4F 5F ED 5E E6 D8 04 :E7
2880 1D 34 06 17 FE 96 32 62 83 00 00 10 27 00 1F AE :1D
2890 44 E6 80 AF 44 1D 34 06 EC 5E 58 49 34 06 58 49 :BA
28A0 58 49 E3 E1 E3 E1 83 00 30 ED 5E 16 FF CF EC 5C :53
28B0 10 27 00 09 EC 5E 40 50 82 00 16 00 02 EC 5E 32 :30
28C0 C4 35 C0 34 40 33 E4 32 7E 4F 5F ED 5E AE 44 E6 :C5
28D0 80 AF 44 1D 10 27 00 0A EC 5E C3 00 01 ED 5E 16 :40
28E0 FF EB EC 5E 35 D0 34 40 33 E4 32 7F 17 F8 C7 17 :62
28F0 03 D4 CC 00 04 ED A9 08 F9 4F 5F ED A9 08 F7 17 :98
-----------------------------------------------------------
Sum: C3 7C 9C D1 C5 CA 6A 30 19 DE D9 B8 DD E2 F1 7A :87

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2900 09 AE CC 00 0F ED A9 08 F5 EC A9 08 F1 34 06 EC :D9
2910 A9 08 F3 34 06 17 28 B1 32 64 CC 00 50 34 06 EC :A6
2920 A9 08 F1 35 10 17 F7 39 34 06 30 A9 01 71 1F 10 :E2
2930 E3 E1 AE A9 08 F3 E6 8B 1D 34 06 17 29 47 32 62 :F9
2940 CC 00 07 ED A9 08 F5 EC A9 08 F1 34 06 EC A9 08 :CB
2950 F3 34 06 17 28 73 32 64 17 F9 74 E7 5F CC 00 50 :5B
2960 34 06 EC A9 08 F1 35 10 17 F6 F6 34 06 30 A9 01 :24
2970 71 1F 10 E3 E1 AE A9 08 F3 E6 8B 1D 34 06 17 29 :BE
2980 04 32 62 E6 5F 1D 34 06 17 FD 91 32 62 83 00 00 :F0
2990 10 27 00 3A EC A9 08 F7 83 00 00 10 26 00 10 E6 :B4
29A0 5F 1D 83 00 30 10 26 00 06 17 04 51 16 FF 50 E6 :22
29B0 5F 1D 34 06 EC A9 08 F7 58 49 34 06 58 49 58 49 :67
29C0 E3 E1 E3 E1 83 00 30 ED A9 08 F7 16 FF 31 E6 5F :5B
29D0 1D 10 83 00 1E 27 08 10 83 00 6B 10 26 00 06 17 :4E
29E0 09 44 16 02 D6 10 83 00 1F 27 08 10 83 00 6A 10 :29
29F0 26 00 06 17 09 40 16 02 C2 10 83 00 20 27 0E 10 :5E
-----------------------------------------------------------
Sum: A3 C0 02 C2 CE 1E EE D8 47 03 47 03 C8 31 E2 77 :BF
```

リスト2　vimマシン語ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2A00 83 00 1C 27 08 10 83 00 6C 10 26 00 06 17 09 5B :84
2A10 16 02 A8 10 83 00 08 27 0E 10 83 00 1D 27 08 10 :7F
2A20 83 00 68 10 26 00 06 17 09 AC 16 02 8E 10 83 00 :2C
2A30 2B 27 08 10 83 00 0A 10 26 00 06 17 09 DB 16 02 :46
2A40 7A 10 83 00 2D 10 26 00 06 17 08 CE 16 02 6C 10 :F7
2A50 83 00 24 10 26 00 06 17 03 F2 16 02 5E 10 83 00 :F8
2A60 77 10 26 00 06 17 19 E9 16 02 50 10 83 00 62 10 :39
2A70 26 00 06 17 1D E8 16 02 42 10 83 00 65 10 26 00 :D0
2A80 06 17 1B 3F 16 02 34 10 83 00 57 10 26 00 06 17 :00
2A90 1A 54 16 02 26 10 83 00 42 10 26 00 06 17 1E D4 :C6
2AA0 16 02 18 10 83 00 45 10 26 00 06 17 1B AA 16 02 :38
2AB0 0A 10 83 00 06 10 26 00 06 17 0A F0 16 01 FC 10 :13
2AC0 83 00 04 10 26 00 0D CC 00 19 34 06 17 0B 0A 32 :47
2AD0 62 16 01 E7 10 83 00 02 10 26 00 06 17 0A E7 16 :4F
2AE0 01 D9 10 83 00 15 10 26 00 0D CC FF FF 34 06 17 :E0
2AF0 0A E7 32 62 16 01 C4 10 83 00 47 10 26 00 23 EC :7F
---------------------------------------------------------
Sum: 11 9C 1A AB BB DA F9 74 8E 5A 8A 2B C6 56 71 D5 :73
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2B00 A9 08 F7 10 27 00 0E EC A9 08 F7 34 06 17 17 E9 :D2
2B10 32 62 16 00 0A 17 26 54 34 06 17 17 DC 32 62 16 :33
2B20 01 99 10 83 00 63 10 26 00 06 17 0B B3 16 01 8B :43
2B30 10 83 00 0B 27 08 10 83 00 48 10 26 00 0C 4F 5F :98
2B40 34 06 17 02 F8 32 62 16 01 71 10 83 00 4C 10 26 :7C
2B50 00 0D CC 00 17 34 06 17 02 E3 32 62 16 01 5C 10 :3D
2B60 83 00 4D 10 26 00 0D CC 00 0C 34 06 17 02 CE 32 :3E
2B70 62 16 01 47 10 83 00 3A 10 26 00 06 17 1F 7D 16 :92
2B80 01 39 10 83 00 2F 10 26 00 06 17 13 67 16 01 2B :0B
2B90 10 83 00 6E 10 26 00 06 17 13 AA 16 01 1D 10 83 :D8
2BA0 00 4E 10 26 00 06 17 13 AD 16 01 0F 10 83 00 4A :64
2BB0 10 26 00 06 17 11 8D 16 01 01 10 83 00 78 10 26 :4A
2BC0 00 06 17 09 CB 16 00 F3 10 83 00 61 10 26 00 06 :2A
2BD0 17 0A 8E 16 00 E5 10 83 00 41 10 26 00 06 17 0A :DB
2BE0 BE 16 00 D7 10 83 00 69 10 26 00 0D CC 00 01 34 :EB
2BF0 06 17 0B F7 32 62 16 00 C2 10 83 00 49 10 26 00 :9D
---------------------------------------------------------
Sum: 01 1C 1E 01 D1 B7 A3 50 97 0C 10 BC 76 43 DF C9 :87
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2C00 10 17 01 F9 CC 00 01 34 06 17 0B DF 32 62 16 00 :D3
2C10 AA 10 83 00 52 10 26 00 0C 4F 5F 34 06 17 0B CB :A6
2C20 32 62 16 00 96 10 83 00 72 10 26 00 06 17 08 07 :A7
2C30 16 00 88 10 83 00 6F 10 26 00 06 17 0D 21 16 00 :37
2C40 7A 10 83 00 4F 10 26 00 06 17 0D 2D 16 00 6C 10 :7B
2C50 83 00 64 10 26 00 06 17 0F AD 16 00 5E 10 83 00 :FD
2C60 44 10 26 00 06 17 12 3B 16 00 50 10 83 00 3E 10 :2B
2C70 26 00 06 17 1D BA 16 00 42 10 83 00 3C 10 26 00 :77
2C80 06 17 1E 21 16 00 34 10 83 00 3F 10 26 00 06 17 :CB
2C90 17 23 16 00 26 10 83 00 43 10 26 00 06 17 08 0D :B4
2CA0 16 00 18 10 83 00 5A 10 26 00 06 17 02 E8 16 00 :6E
2CB0 0A CC 00 07 34 06 17 F6 25 32 62 4F 5F ED A9 08 :29
2CC0 F7 16 FC 3B 35 C2 34 40 33 E4 32 76 CC 00 0C 34 :7A
2CD0 06 17 F6 0A 32 62 CC 80 00 ED A9 01 67 CC F7 FF :BD
2CE0 ED A9 01 69 CC 6F FC 1F 01 EC 84 ED A9 01 6B 4F :18
2CF0 5F ED A9 08 F3 ED A9 08 F1 CC 00 07 ED A9 08 F5 :E5
---------------------------------------------------------
Sum: EF 72 1D 1E E8 97 3A 93 4D 15 B8 48 D4 33 D5 95 :BB
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2D00 EC A9 01 69 ED A9 01 6D EC A9 01 67 34 06 17 14 :65
2D10 E7 32 62 32 C4 35 C0 34 40 33 E4 32 7A 4F 5F ED :38
2D20 5E EC 5E 83 00 18 10 2C 00 CF EC A9 01 6D A3 A9 :9D
2D30 01 69 10 27 00 7B 4F 5F ED 5C AE A9 01 6D E6 80 :3E
2D40 AF A9 01 6D 1D 34 06 EC 5C C3 00 01 ED 5C 83 00 :F5
2D50 01 34 06 CC 00 50 34 06 EC 5E 35 10 17 F3 02 34 :60
2D60 06 30 A9 01 71 1F 10 E3 E1 E3 E1 1F 01 35 06 E7 :4A
2D70 84 83 00 00 10 26 FF C2 EC 5C 83 00 50 10 2C 00 :55
2D80 2D EC 5C C3 00 01 ED 5C 83 00 01 34 06 CC 00 50 :5C
2D90 34 06 EC 5E 35 10 17 F2 C8 34 06 30 A9 01 71 1F :3E
2DA0 10 E3 E1 E3 E1 1F 01 4F 5F E7 84 16 FF CA 16 00 :C6
2DB0 35 4F 5F ED 5C EC 5C 83 00 50 10 2C 00 28 CC 00 :77
2DC0 50 34 06 EC 5E 35 10 17 F2 97 34 06 30 A9 01 71 :3E
2DD0 1F 10 E3 E1 AE 5C 30 8B 4F 5F E7 84 EC 5C C3 00 :DC
2DE0 01 ED 5C 16 FF CF EC 5E 34 06 17 24 C9 32 62 EC :36
2DF0 5E C3 00 01 ED 5E 16 FF 28 32 C4 35 C0 34 40 33 :3C
---------------------------------------------------------
Sum: E0 D8 4E 54 B9 14 0C E2 75 00 A9 A4 58 ED 6F 44 :CF
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2E00 E4 4F 5F ED A9 08 F3 CC 00 50 34 06 EC A9 08 F1 :07
2E10 35 10 17 F2 4C 34 06 30 A9 01 71 1F 10 E3 E1 AE :C0
2E20 A9 08 F3 E6 8B 1D 83 00 20 10 26 00 0E EC A9 08 :B6
2E30 F3 C3 00 01 ED A9 08 F3 16 FF CC 35 C0 34 40 33 :C5
2E40 E4 EC 44 ED A9 08 F1 17 FF B3 35 C0 34 40 33 E4 :EC
2E50 4F 5F ED A9 08 F3 CC 00 50 34 06 EC A9 08 F1 35 :58
2E60 10 17 F1 FD 34 06 30 A9 01 71 1F 10 E3 E1 AE A9 :E4
2E70 08 F3 E6 8B 1D 10 27 00 0E EC A9 08 F3 C3 00 01 :22
2E80 ED A9 08 F3 16 FF CF 35 C0 34 40 33 E4 32 7C EC :8F
2E90 46 83 00 18 10 26 00 37 EC 46 34 06 EC 44 34 06 :24
2EA0 17 23 26 32 64 4F 5F ED 5E CC 00 50 A3 44 83 00 :75
2EB0 01 34 06 EC 5E A3 E1 10 2C 00 14 CC 00 20 34 06 :7F
2EC0 17 F4 1B 32 62 EC 5E C3 00 01 ED 5E 16 FF DA 4F :51
2ED0 5F ED 5E EC 46 34 06 EC 5E E3 44 34 06 17 22 E9 :E3
2EE0 32 64 17 F3 EA ED 5C 83 00 08 10 26 00 2D EC 5E :0B
2EF0 83 00 00 10 2F 00 21 EC 46 34 06 EC 5E C3 FF FF :5A
---------------------------------------------------------
Sum: 76 47 35 2E 18 37 88 36 17 0A 69 17 6A 78 F2 2A :CC
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2F00 ED 5E 83 FF FF E3 44 34 06 17 22 BD 32 64 CC 00 :85
2F10 20 34 06 17 F3 C8 32 62 16 00 3E EC 5C 83 00 20 :FF
2F20 10 2D 00 35 EC 5C 83 00 80 10 2C 00 2C EC 5E 30 :9F
2F30 A9 08 FB 30 8B EC 5C E7 84 EC 46 34 06 EC 5E C3 :93
2F40 00 01 ED 5E 83 00 01 E3 44 34 06 17 22 7B 32 64 :7B
2F50 EC 5C 34 06 17 F3 87 32 62 EC 5C 83 00 0A 10 27 :B3
2F60 00 09 EC 5C 83 00 1B 10 26 FF 68 EC 46 34 06 EC :E4
2F70 5E E3 44 34 06 17 22 51 32 64 CC 00 20 34 06 17 :1C
2F80 F3 5C 32 62 EC 5E 30 A9 08 FB 30 8B 4F 5F E7 84 :DD
2F90 EC 5C 32 C4 35 C0 34 40 33 E4 32 7E 17 F3 30 83 :2B
2FA0 00 5A 10 26 00 22 17 1E 8D CC 6F FC 1F 01 EC A9 :60
2FB0 01 6B ED 84 CC 00 18 34 06 4F 5F 34 06 17 22 09 :25
2FC0 32 64 17 F1 ED 17 F0 58 35 C6 34 40 33 E4 35 C0 :65
2FD0 34 40 33 E4 32 7C 4F 5F ED 5E EC 5E 83 00 18 10 :27
2FE0 2C 00 46 4F 5F ED 5C CC 00 50 34 06 EC 5E 35 10 :4E
2FF0 17 F0 6E 34 06 30 A9 01 71 1F 10 E3 E1 34 06 EC :13
---------------------------------------------------------
Sum: 99 21 34 97 FD ED F1 B2 7F 23 FC 23 56 8C 83 26 :5E
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3000 5C C3 00 01 ED 5C 83 00 01 35 10 E6 8B 1D AE A9 :17
3010 01 6B E7 80 AF A9 01 6B 83 00 00 10 26 FF C8 EC :03
3020 5E C3 00 01 ED 5E 16 FF B1 35 D6 34 40 33 E4 32 :FB
3030 78 EC A9 08 F1 83 00 00 10 2C 00 4F EC A9 01 6B :15
3040 A3 A9 01 67 10 26 00 17 CC 00 07 34 06 17 F2 8E :A5
3050 32 62 4F 5F ED A9 08 F3 ED A9 08 F1 16 00 2C 17 :BB
3060 01 F1 CC 00 17 34 06 17 0B 5D 32 62 17 00 C3 CC :C8
3070 00 17 34 06 17 22 EE 32 62 17 21 A1 4F 5F 34 06 :CD
3080 17 22 33 32 62 4F 5F ED A9 08 F1 EC A9 08 F1 83 :4E
3090 00 17 10 2F 00 29 EC A9 01 6D A3 A9 01 69 10 26 :6E
30A0 00 1A CC 00 07 34 06 17 F2 34 32 62 CC 00 17 ED :C8
30B0 A9 08 F1 4F 5F ED A9 08 F3 16 00 03 17 00 2A CC :07
30C0 00 50 34 06 EC A9 08 F1 35 10 17 EF 94 34 06 30 :61
30D0 A9 01 71 1F 10 E3 E1 AE A9 08 F3 E6 8B 1D 10 26 :24
30E0 00 03 17 FD 67 32 C4 35 C0 34 40 33 E4 17 00 AF :BA
30F0 CC 00 17 34 06 17 0A F1 32 62 17 00 D0 CC 00 18 :8E
---------------------------------------------------------
Sum: 3E 9F B3 5C D6 79 47 37 CA 20 6F A3 BF 13 C8 28 :77
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3100 34 06 4F 5F 34 06 17 20 C0 32 64 30 8C 02 20 02 :8F
3110 0A 00 34 10 17 F1 DB 32 62 CC 00 17 34 06 17 21 :1A
3120 95 32 62 CC 00 17 ED A9 08 F1 4F 5F ED A9 08 F3 :DA
3130 35 C0 34 40 33 E4 32 7C EC A9 01 6B A3 A9 01 67 :E3
3140 10 27 00 59 EC A9 01 6B 34 06 17 04 D1 32 62 ED :38
3150 A9 01 6B ED 5C 4F 5F ED 5E AE 5C E6 80 AF 5C 1D :EF
3160 34 06 EC 5E C3 00 01 ED 5E 83 00 01 30 A9 01 71 :62
3170 30 8B 35 06 E7 84 83 00 00 10 26 FF DC EC 5E 83 :C2
3180 00 50 10 2C 00 17 EC 5E C3 00 01 ED 5E 83 00 01 :80
3190 30 A9 01 71 30 8B 4F 5F E7 84 16 FF E0 35 D6 34 :53
31A0 40 33 E4 32 7E 4F 5F ED 5E EC 5E C3 00 01 ED 5E :59
31B0 83 00 01 30 A9 01 71 E6 8B 1D AE A9 01 6B E7 80 :87
31C0 AF A9 01 6B 83 00 00 10 26 FF DE 35 C6 34 40 33 :FC
31D0 E4 32 7E EC A9 01 6D A3 A9 01 69 10 27 00 4F 4F :22
31E0 5F ED 5E AE A9 01 6D E6 80 AF A9 01 6D 1D 34 06 :F2
31F0 EC 5E C3 00 01 ED 5E 83 00 01 30 A9 08 A1 30 8B :1A
---------------------------------------------------------
Sum: F6 03 3B 29 9D 4F 38 68 E8 1C 90 42 4E E6 FA A1 :8E
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3200 35 06 E7 84 83 00 00 10 26 FF D8 EC 5E 83 00 50 :53
3210 10 2C 00 17 EC 5E C3 00 01 ED 5E 83 00 01 30 A9 :09
3220 08 A1 30 8B 4F 5F E7 84 16 FF E0 16 00 23 4F 5F :59
3230 ED 5E EC 5E 83 00 50 10 2C 00 16 EC 5E 30 A9 08 :E5
3240 A1 30 8B 4F 5F E7 84 EC 5E C3 00 01 ED 5E 16 FF :E3
3250 E1 35 C6 34 40 33 E4 32 7C EC A9 01 6D A3 A9 01 :65
3260 69 10 26 00 0E E6 A9 08 A1 1D 83 00 00 10 26 00 :BB
3270 02 35 D6 30 A9 08 A1 34 10 17 00 7A 32 62 34 06 :32
3280 EC A9 01 6D A3 E1 ED A9 01 6D ED 5C 4F 5F ED 5E :CD
3290 EC 5E C3 00 01 ED 5E 83 00 01 30 A9 08 A1 E6 8B :D0
32A0 1D AE 5C E7 80 AF 5C 83 00 00 10 26 FF E2 35 D6 :3E
32B0 34 40 33 E4 EC A9 01 6D A3 A9 01 6B 83 01 00 10 :DA
32C0 2C 00 31 CC 00 18 34 06 4F 5F 34 06 17 1E FA 32 :C4
32D0 64 30 8C 02 20 17 07 6E 65 61 72 6C 79 20 4D 65 :BD
32E0 6D 6F 72 79 20 66 75 6C 6C 20 21 21 00 34 10 17 :57
32F0 F0 00 32 62 35 C0 34 40 33 E4 32 7E CC 00 01 ED :6E
---------------------------------------------------------
Sum: 3D 6F 04 18 1C 40 38 3A EB A9 7F 94 7D 9F A1 D0 :CA
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3300 5E AE 44 E6 80 AF 44 1D 10 27 00 0A EC 5E C3 00 :14
3310 01 ED 5E 16 FF EB EC 5E 35 D0 34 40 33 E4 17 00 :3D
3320 05 17 FA D9 35 C0 34 40 33 E4 CC FF FF 34 06 17 :8A
3330 00 14 32 62 35 C0 34 40 33 E4 CC 00 01 34 06 17 :46
3340 00 04 32 62 35 C0 34 40 33 E4 EC A9 08 F1 E3 44 :CD
3350 ED A9 08 F1 17 FC D4 EC A9 08 F7 C3 FF FF ED A9 :61
3360 08 F7 83 00 00 10 2E FF E1 35 C0 34 40 33 E4 CC :EC
3370 00 50 34 06 EC A9 08 F1 35 10 17 EC E4 34 06 30 :AE
3380 A9 01 71 1F 10 E3 E1 AE A9 08 F3 E6 8B 1D 83 00 :71
3390 00 10 27 00 0B EC A9 08 F3 83 00 4F 10 26 00 14 :EE
33A0 EC A9 08 F1 C3 00 01 ED A9 08 F1 4F 5F ED A9 08 :2D
33B0 F3 16 00 0B EC A9 08 F3 C3 00 01 ED A9 08 F3 17 :10
33C0 FC 69 EC A9 08 F7 C3 FF FF ED A9 08 F7 83 00 00 :D2
33D0 10 2E FF 9B 35 C0 34 40 33 E4 EC A9 08 F3 83 00 :6B
33E0 01 ED A9 08 F3 EC A9 08 F3 83 00 00 10 2C 00 12 :F3
33F0 CC 00 4F ED A9 08 F3 EC A9 08 F1 83 00 01 ED A9 :54
---------------------------------------------------------
Sum: BA 0E 42 E4 C4 B2 FC E0 73 DF F1 7A FC DC 2F 05 :09
```

リスト2　vimマシン語ダンプ・リスト

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3400  08 F1 17 FC 26 EC A9 08 F7 C3 FF FF ED A9 08 F7 :1C
3410  83 00 00 10 2E FF C3 35 C0 34 40 33 E4 17 FF 16 :2F
3420  17 F9 DA EC A9 08 F7 C3 FF FF ED A9 08 F7 83 00 :57
3430  00 10 2E FF E8 35 C0 34 40 33 E4 32 7F EC A9 08 :F3
3440  F3 83 00 4F 10 27 00 63 EC A9 08 F1 34 06 EC A9 :BC
3450  08 F3 34 06 17 1D 72 32 64 17 EE 73 E7 5F 83 00 :B2
3460  20 10 2D 00 46 E6 5F 1D 83 00 7F 10 2C 00 3C EC :6B
3470  A9 08 F3 C3 00 01 ED A9 08 F3 83 00 01 34 06 CC :83
3480  00 50 34 06 EC A9 08 F1 35 10 17 EB D4 34 06 30 :9D
3490  A9 01 71 1F 10 E3 E1 E3 E1 1F 01 E6 5F 1D E7 84 :BF
34A0  EC A9 08 F1 34 06 17 1E 0D 32 62 35 C2 34 40 33 :3C
34B0  E4 CC 00 18 34 06 4F 5F 34 06 17 1D 0C 32 64 30 :F0
34C0  8C 02 20 2D 43 6F 70 79 72 69 67 68 74 20 28 63 :3F
34D0  29 20 31 39 38 33 20 62 79 20 4E 2E 54 73 75 64 :55
34E0  61 2C 20 54 73 75 6B 75 62 61 20 55 6E 69 76 2E :7C
34F0  00 34 10 17 ED FC 32 62 35 C0 34 40 33 E4 32 7E :08
---------------------------------------------------------
Sum:  F5 D0 A1 0E 91 FE 5D 92 AA ED A2 CF 0A D3 BA 00 :91
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3500  EC A9 08 F3 ED 5E EC 5E 83 00 4F 10 2C 00 50 CC :4F
3510  00 50 34 06 EC A9 08 F1 35 10 17 EB 44 34 06 30 :0D
3520  A9 01 71 1F 10 E3 E1 34 06 EC 5E C3 00 01 35 10 :9B
3530  E6 8B 1D 34 06 CC 00 50 34 06 EC A9 08 F1 35 10 :F1
3540  17 EB 1E 34 06 30 A9 01 71 1F 10 E3 E1 AE 5E 30 :D4
3550  8B 35 06 E7 84 EC 5E C3 00 01 ED 5E 16 FF A7 CC :12
3560  00 50 34 06 EC A9 08 F1 35 10 17 EA F4 34 06 30 :BC
3570  A9 01 71 1F 10 E3 E1 C3 00 4F 1F 01 4F 5F E7 84 :59
3580  EC 44 C3 FF FF ED 44 83 00 00 10 2E FF 72 35 C6 :4F
3590  34 40 33 E4 EC A9 08 F7 34 06 17 FF 5D 32 62 EC :4C
35A0  A9 08 F1 34 06 17 1D 0E 32 62 35 C0 34 40 33 E4 :32
35B0  CC 00 19 34 06 17 00 21 32 62 CC 00 19 34 06 17 :21
35C0  00 17 32 62 35 C0 34 40 33 E4 17 0E 44 83 00 18 :2F
35D0  34 06 17 0D 24 32 62 35 C0 34 40 33 E4 32 7A EC :2E
35E0  A9 08 F1 ED 5C EC A9 08 F3 ED 5A 4F 5F ED 5E EC :A7
35F0  5E 83 00 0C 10 2C 00 13 EC 44 ED A9 08 F1 17 FA :0C
---------------------------------------------------------
Sum:  96 2A CD 3F 31 2C 6D 84 02 94 A9 B9 EA 11 71 63 :E1
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3600  2A EC 5E C3 00 01 ED 5E 16 FF E4 EC 5C ED A9 08 :62
3610  F1 EC 5A ED A9 08 F3 17 FA 11 32 C4 35 C0 34 40 :49
3620  33 E4 EC 44 A3 A9 01 67 10 27 00 31 EC 44 A3 A9 :DF
3630  01 6D 10 27 00 27 EC 44 C3 FF FF ED 44 A3 A9 01 :3B
3640  67 10 27 00 18 EC 44 A3 A9 01 6D 10 27 00 0E EC :D1
3650  44 83 00 01 1F 01 E6 84 1D 10 26 FF D9 EC 44 35 :E2
3660  C0 34 40 33 E4 CC 00 50 34 06 EC A9 08 F1 35 10 :74
3670  17 E9 EE 34 06 30 A9 01 71 1F 10 E3 E1 AE A9 08 :C5
3680  F3 E6 8B 1D 10 27 00 0B EC A9 08 F3 C3 00 01 ED :04
3690  A9 08 F3 CC 00 01 34 06 17 01 50 32 62 35 C0 34 :D0
36A0  40 33 E4 CC 00 50 34 06 EC A9 08 F1 35 10 17 E9 :80
36B0  B0 34 06 30 A9 01 71 1F 10 E3 E1 AE A9 08 F3 E6 :60
36C0  8B 1D 10 27 00 0E EC A9 08 F3 C3 00 01 ED A9 08 :DF
36D0  F3 16 FF CF CC 00 01 34 06 17 01 0F 32 62 35 C0 :8E
36E0  34 40 33 E4 17 EB E8 83 00 77 10 26 00 18 17 05 :D9
36F0  6D EC A9 08 F1 34 06 17 1B BC 32 62 CC 00 01 34 :B8
---------------------------------------------------------
Sum:  7C 8D 5C 4A FA 68 54 45 76 DF EB C4 AC D3 1A 1C :63
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3700  06 17 00 E7 32 62 35 C0 34 40 33 E4 32 7E E6 47 :F5
3710  10 27 00 40 EC A9 08 F3 83 00 4F 10 2C 00 26 CC :07
3720  00 50 34 06 EC A9 08 F1 35 10 17 E9 34 34 06 30 :FB
3730  A9 01 71 1F 10 E3 E1 34 06 CC 00 4E 35 10 E6 8B :18
3740  1D 10 27 00 0F CC 00 07 34 06 17 EB 91 32 62 CC :63
3750  00 01 35 D0 E6 47 10 27 00 5F CC 00 4F ED 5E EC :1B
3760  5E A3 A9 08 F3 10 2F 00 50 CC 00 50 34 06 EC A9 :1F
3770  08 F1 35 10 17 E8 EA 34 06 30 A9 01 71 1F 10 E3 :BE
3780  E1 34 06 EC 5E 83 00 01 35 10 E6 8B 1D 34 06 CC :C2
3790  00 50 34 06 EC A9 08 F1 35 10 17 E8 C4 34 06 30 :8A
37A0  A9 01 71 1F 10 E3 E1 AE 5E 30 8B 35 06 E7 84 EC :67
37B0  5E C3 FF FF ED 5E 16 FF A6 CC 00 50 34 06 EC A9 :10
37C0  08 F1 35 10 17 E8 9A 34 06 30 A9 01 71 1F 10 E3 :6E
37D0  E1 AE A9 08 F3 30 8B E6 45 1D E7 84 EC A9 08 F3 :31
37E0  C3 00 01 ED A9 08 F3 4F 5F 35 D0 34 40 33 E4 32 :C5
37F0  7B EC A9 08 F1 34 06 EC A9 08 F3 34 06 17 19 C9 :06
---------------------------------------------------------
Sum:  51 07 11 51 04 63 6C 2E 3D 23 00 4C 0A 6D 45 74 :97
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3800  32 64 17 EA CA E7 5F E6 5F 1D 83 00 20 10 2D 00 :E9
3810  0A E6 5F 1D 83 00 7E 10 2F 00 0A E6 5F 1D 83 00 :9B
3820  00 10 2C 00 1C EC 44 34 06 E6 5F 1D 34 06 17 FE :73
3830  D7 32 64 EC A9 08 F1 34 06 17 1A 7A 32 62 16 FF :89
3840  B0 E6 5F 1D 83 00 09 10 26 00 3C EC 44 34 06 EC :46
3850  00 20 34 06 17 FE B1 32 64 83 00 00 10 27 00 03 :73
3860  16 00 16 EC A9 08 F9 34 06 EC A9 08 F3 35 10 17 :E8
3870  E8 34 83 00 00 10 26 FF D2 EC A9 08 F1 34 06 17 :85
3880  1A 34 32 62 16 FF 6A E6 5F 1D 83 00 0A 10 26 00 :86
3890  0C 4F 5F 34 06 17 00 FB 32 62 16 FF 54 E6 5F 1D :65
38A0  83 00 08 10 26 00 A4 EC A9 08 F3 83 00 00 10 2F :B7
38B0  00 96 EC A9 08 F3 ED 5D EC 5D 83 00 50 10 2C 00 :C8
38C0  50 CC 00 50 34 06 EC A9 08 F1 35 10 17 E7 92 34 :3D
38D0  06 30 A9 01 71 1F 10 E3 E1 AE 5D E6 8B 1D 34 06 :17
38E0  EC 5D 83 00 01 34 06 CC 00 50 34 06 EC A9 08 F1 :EB
38F0  35 10 17 E7 6C 34 06 30 A9 01 71 1F 10 E3 E1 E3 :0A
---------------------------------------------------------
Sum:  E1 48 FA 89 B1 87 EE 85 B4 49 DA 16 69 EF 69 54 :59
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3900  E1 1F 01 35 06 E7 84 EC 5D C3 00 01 ED 5D 16 FF :13
3910  A7 CC 00 50 34 06 EC A9 08 F1 35 10 17 E7 42 34 :44
3920  06 30 A9 01 71 1F 10 E3 E1 C3 00 4F 1F 01 4F 5F :24
3930  E7 84 EC A9 08 F3 C3 FF FF ED A9 08 F3 EC A9 08 :EA
3940  F1 34 06 17 19 70 32 62 16 FE A6 E6 5F 1D 83 00 :FE
3950  1B 10 26 00 03 16 00 03 16 FE 96 32 C4 35 C0 34 :36
3960  40 33 E4 CC 00 02 34 06 17 00 28 32 62 CC 00 01 :FF
3970  34 06 17 FE 76 32 62 35 C0 34 40 33 E4 CC 00 01 :A6
3980  34 06 17 00 0E 32 62 CC 00 01 34 06 17 FE 5C 32 :9D
3990  62 35 C0 34 40 33 E4 32 7C EC 44 83 00 01 10 26 :7A
39A0  00 1F 17 F8 AE EC A9 08 F1 83 00 17 10 27 00 0E :49
39B0  CC 00 17 A3 A9 08 F1 34 06 17 02 0B 32 62 16 00 :30
39C0  5D EC A9 08 F1 83 00 17 10 27 00 29 17 F8 84 EC :64
39D0  A9 08 F1 C3 00 01 ED A9 08 F1 83 00 01 83 00 16 :12
39E0  10 27 00 0E CC 00 17 A3 A9 08 F1 34 06 17 01 D7 :96
39F0  32 62 16 00 29 17 F7 A7 CC 00 17 34 06 17 01 E9 :A6
---------------------------------------------------------
Sum:  9F F3 72 B8 D0 AD E6 5B 48 3B 87 21 FC 4C 9B F8 :80
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3A00  32 62 CC 00 18 34 06 4F 5F 34 06 17 17 BB 32 64 :19
3A10  30 8C 02 20 02 0A 00 34 10 17 E8 D6 32 62 4F 5F :45
3A20  ED 5E EC 5E 83 00 50 10 2C 00 2A CC 00 50 34 06 :24
3A30  EC A9 08 F1 35 10 17 E6 28 34 06 30 A9 01 71 1F :9C
3A40  10 E3 E1 AE 5E 30 8B 4F 5F E7 84 EC 5E C3 00 01 :C2
3A50  ED 5E 16 FF CD EC A9 08 F3 ED 5E 4F 5F ED 5C ED :EC
3A60  A9 08 F3 EC A9 08 F1 83 00 00 10 27 01 0E CC 00 :C7
3A70  50 34 06 EC A9 08 F1 83 00 01 35 10 17 E5 E2 34 :F3
3A80  06 30 A9 01 71 1F 10 E3 E1 AE A9 08 F3 E6 8B 1D :24
3A90  83 00 20 10 26 00 34 EC A9 08 F3 C3 00 01 ED A9 :F7
3AA0  08 F3 83 00 01 34 06 CC 00 50 34 06 EC A9 08 F1 :9D
3AB0  35 10 17 E5 AC 34 06 30 A9 01 71 1F 10 E3 E1 E3 :48
3AC0  E1 1F 01 CC 00 20 E7 84 16 FF A3 EC 44 83 00 00 :C3
3AD0  10 26 00 A8 CC 00 50 34 06 EC A9 08 F1 83 00 01 :46
3AE0  35 10 17 E5 7C 34 06 30 A9 01 71 1F 10 E3 E1 AE :E3
3AF0  5E E6 8B 1D 10 27 00 84 CC 00 50 34 06 EC A9 08 :9A
---------------------------------------------------------
Sum:  7B E0 B8 60 EB 7C 10 0D D9 47 93 92 01 59 1B 5B :0C
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3B00  F1 83 00 01 35 10 17 E5 58 34 06 30 A9 01 71 1F :B2
3B10  10 E3 E1 AE 5E E6 8B 1D 34 06 EC 5C C3 00 01 ED :A1
3B20  5C 83 00 01 E3 A9 08 F3 34 06 CC 00 50 34 06 EC :E3
3B30  A9 08 F1 35 10 17 E5 29 34 06 30 A9 01 71 1F 10 :C0
3B40  E3 E1 E3 E1 1F 01 35 06 E7 84 EC 5E C3 00 01 ED :49
3B50  5E 83 00 01 34 06 CC 00 50 34 06 EC A9 08 F1 83 :83
3B60  00 01 35 10 17 E4 FA 34 06 30 A9 01 71 1F 10 E3 :D2
3B70  E1 E3 E1 1F 01 4F 5F E7 84 16 FF 58 EC A9 08 F1 :D9
3B80  83 00 00 10 26 00 19 CC 00 17 34 06 17 17 D6 32 :25
3B90  62 17 16 89 4F 5F 34 06 17 17 CA 32 62 16 00 25 :C7
3BA0  EC A9 08 F1 83 00 01 ED 5E EC 5E 83 00 18 10 2C :7E
3BB0  00 13 EC 5E 34 06 17 16 FD 32 62 EC 5E C3 00 01 :63
3BC0  ED 5E 16 FF E4 35 D6 34 40 33 A9 01 71 33 C9 07 :14
3BD0  30 30 C8 50 A6 65 C6 28 3D ED E3 EC C3 ED 83 6A :07
3BE0  61 26 F8 6A E4 2A F4 35 C6 34 40 30 A9 01 71 33 :D8
3BF0  88 50 A6 65 C6 28 3D ED E3 EC C1 ED 81 6A 61 26 :EA
---------------------------------------------------------
Sum:  FF 10 51 FC 51 41 1B 92 4D D0 D3 89 BB 09 A5 9A :17
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3C00  F8 6A E4 2A F4 35 C6 34 40 33 E4 32 7F 17 E6 BF :57
3C10  E7 5F 83 00 64 10 26 00 19 EC A9 08 F7 34 06 EC :36
3C20  A9 08 F1 34 06 17 00 85 32 64 4F 5F ED A9 08 F3 :4D
3C30  35 C2 E6 5F 1D 83 00 77 10 26 00 20 17 00 1F EC :CB
3C40  A9 08 F7 C3 FF FF ED A9 08 F7 83 00 00 10 2E FF :BE
3C50  EB EC A9 08 F1 34 06 17 16 5C 32 62 35 C2 34 40 :3B
3C60  33 E4 32 7F CC 00 50 34 06 EC A9 08 F1 35 10 17 :08
3C70  E3 EF 34 06 30 A9 01 71 1F 10 E3 E1 AE A9 08 F3 :9C
3C80  E6 8B 1D 10 27 00 0A E6 5F 1D 83 00 20 10 26 00 :0A
3C90  04 4F 5F 20 03 CC 00 01 E7 5F 83 00 00 10 27 00 :A2
3CA0  0A 17 0A 9C 34 06 17 F8 51 32 62 35 C2 34 40 33 :93
3CB0  E4 32 7A EC 44 ED 5E EC 5E 83 00 17 10 2C 00 4F :7A
3CC0  CC 00 50 34 06 EC 5E 35 10 17 E3 95 34 06 30 A9 :87
3CD0  01 71 1F 10 E3 E1 ED 5A 4F 5F ED 5C EC 5C 83 00 :6E
3CE0  50 10 2C 00 20 EC 5C C3 00 50 AE 5A E6 8B 1D 34 :D1
3CF0  06 EC 5C AE 5A 30 8B 35 06 E7 84 EC 5C C3 00 01 :C3
---------------------------------------------------------
Sum:  62 EA 3B B7 6C 63 E1 E7 38 D6 87 87 A2 D4 EA 33 :84
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3D00  ED 5C 16 FF D7 EC 5E C3 00 01 ED 5E 16 FF A8 17 :62
3D10  F4 BB EC 46 C3 FF FF ED 46 83 00 00 10 2E FF 93 :28
3D20  EC 44 ED 5E EC 5E 83 00 18 10 2C 00 13 EC 5E 34 :2D
3D30  06 17 15 82 32 62 EC 5E C3 00 01 ED 5E 16 FF E4 :9A
3D40  32 C4 35 C0 34 40 33 E4 32 7C EC A9 08 F1 83 00 :35
3D50  17 10 26 00 02 35 D6 CC 00 50 34 06 EC A9 08 F1 :3E
3D60  35 10 17 E2 FC 34 06 30 A9 01 71 1F 10 E3 E1 AE :60
3D70  A9 08 F3 E6 8B 1D 10 27 00 1C EC A9 08 F3 C3 00 :CA
3D80  01 ED A9 08 F3 16 FF CF EC A9 08 F3 ED 5E 4F 5F :FF
3D90  ED 5C CC 00 50 34 06 EC A9 08 F1 C3 00 01 35 10 :36
3DA0  17 E2 BE 34 06 30 A9 01 71 1F 10 E3 E1 AE 5C E6 :1F
3DB0  8B 1D 83 00 20 10 26 00 0A EC 5C C3 00 01 ED 5C :E0
3DC0  16 FF CF EC 5E 83 00 50 10 2C 00 2D EC 5E C3 00 :77
3DD0  01 ED 5E 83 00 01 34 06 CC 00 50 34 06 EC A9 08 :FD
3DE0  F1 35 10 17 E2 7B 34 06 30 A9 01 71 1F 10 E3 E1 :22
3DF0  E3 E1 1F 01 CC 00 20 E7 84 EC 5E 83 00 50 10 2C :94
---------------------------------------------------------
Sum:  75 A8 7B 70 EA FA 47 14 9C EC AB 73 82 57 5F 27 :4C
```

リスト 2　vimマシン語ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3E00 00 81 CC 00 50 34 06 EC A9 08 F1 C3 00 01 35 10 :6E
3E10 17 E2 4E 34 06 30 A9 01 71 1F 10 E3 E1 AE 5C E6 :AF
3E20 8B 1D 10 27 00 5D CC 00 50 34 06 EC A9 08 F1 C3 :E3
3E30 00 01 35 10 17 E2 2A 34 06 30 A9 01 71 1F 10 E3 :00
3E40 E1 34 06 EC 5C C3 00 01 ED 5C 83 00 01 35 10 E6 :1F
3E50 8B 1D 34 06 EC 5E C3 00 01 ED 5E 83 00 01 34 06 :F9
3E60 CC 00 50 34 06 EC A9 08 F1 35 10 17 E1 F3 34 06 :4E
3E70 30 A9 01 71 1F 10 E3 E1 E3 E1 1F 01 35 06 E7 84 :C8
3E80 16 FF 76 EC A9 08 F1 34 06 17 14 2A 32 62 CC 00 :08
3E90 01 34 06 EC A9 08 F1 C3 00 01 34 06 17 FE 0E 32 :1C
3EA0 64 35 D6 34 40 33 E4 32 7E EC A9 08 F3 ED 5E EC :71
3EB0 5E 83 00 50 10 2C 00 2F EC 5E C3 00 01 ED 5E 83 :78
3EC0 00 01 34 06 CC 00 50 34 06 EC A9 08 F1 35 10 17 :7B
3ED0 E1 8F 34 06 30 A9 01 71 1F 10 E3 E1 E3 E1 1F 01 :CC
3EE0 4F 5F E7 84 16 FF C8 EC A9 08 F1 34 06 17 13 C6 :AE
3EF0 32 62 35 C6 34 40 33 E4 CC 00 18 34 06 4F 5F 34 :1A
-------------------------------------------------------------
Sum: 45 B7 C0 B4 C2 17 06 D8 3C 50 09 B7 2F BB 28 C5 :4A
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3F00 06 17 12 C5 32 64 30 8C 02 20 02 2F 00 34 10 17 :F4
3F10 E3 E0 32 62 CC 00 18 34 06 CC 00 01 34 06 17 EF :82
3F20 68 32 64 83 00 0A 10 26 00 0E 30 A9 08 FB 34 10 :EF
3F30 17 00 34 32 62 16 00 0B 30 A9 08 FB 34 10 17 01 :38
3F40 5E 32 62 35 C0 34 40 33 E4 30 A9 08 FB 34 10 17 :A9
3F50 00 15 32 62 35 C0 34 40 33 E4 30 A9 08 FB 34 10 :49
3F60 17 01 3C 32 62 35 C0 34 40 33 E4 32 7E EC 44 34 :7C
3F70 06 17 00 58 32 62 83 00 00 10 27 00 02 35 C6 EC :AC
3F80 44 34 06 17 00 DD 32 62 ED 5E 83 00 00 10 27 00 :0B
3F90 0B EC 5E 34 06 17 02 4E 32 62 35 C6 CC 00 18 34 :9D
3FA0 06 4F 5F 34 06 17 12 21 32 64 EC 44 34 06 30 8C :F4
3FB0 02 20 10 07 25 73 20 63 61 6E 27 74 20 66 69 6E :1B
3FC0 64 2E 00 34 10 17 E3 2A 32 64 35 C6 34 40 33 E4 :16
3FD0 32 7C EC A9 08 F1 ED 5E EC A9 08 F3 ED 5C CC 00 :2C
3FE0 50 34 06 EC 5E 35 10 17 E0 77 34 06 30 A9 01 71 :0C
3FF0 1F 10 E3 E1 34 06 EC 5C C3 00 01 ED 5C 35 10 E6 :AD
-------------------------------------------------------------
Sum: 3F 05 54 2D C4 D0 41 C7 02 10 5B E1 C0 8B A8 C7 :69
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4000 8B 1D 83 00 00 10 26 00 18 EC 5E C3 00 01 ED 5E :D2
4010 83 00 18 10 26 00 06 4F 5F 32 C4 35 C0 4F 5F ED :0B
4020 5C CC 00 50 34 06 EC 5E 35 10 17 E0 34 34 06 30 :D6
4030 A9 01 71 1F 10 E3 E1 E3 5C 34 06 EC 44 34 06 17 :08
4040 02 89 32 64 83 00 00 10 27 00 13 EC 5E ED A9 08 :D6
4050 F1 EC 5C ED A9 08 F3 CC 00 01 32 C4 35 C0 16 FF :97
4060 7D 35 D6 34 40 33 E4 32 7E EC A9 01 6D ED 5E EC :FD
4070 5E A3 A9 01 69 10 24 00 22 EC 5E 34 06 EC 44 34 :52
4080 06 17 02 47 32 64 83 00 00 10 27 00 04 EC 5E 35 :39
4090 D0 EC 5E C3 00 01 ED 5E 16 FF D4 4F 5F 35 D0 34 :F9
40A0 40 33 E4 32 7E EC 44 34 06 17 00 58 32 62 83 00 :F7
40B0 00 10 27 00 02 35 C6 EC 44 34 06 17 00 E9 32 62 :32
40C0 ED 5E 83 00 00 10 27 00 0B EC 5E 34 06 17 01 16 :C2
40D0 32 62 35 C6 CC 00 18 34 06 4F 5F 34 06 17 10 E9 :A5
40E0 32 64 EC 44 34 06 30 8C 02 20 10 07 25 73 20 63 :10
40F0 61 6E 27 74 20 66 69 6E 64 2E 00 34 10 17 E1 F2 :87
-------------------------------------------------------------
Sum: A9 0F 4F BF 11 46 46 4A A6 1E 59 0A 14 62 AE D8 :D0
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4100 32 64 35 C6 34 40 33 E4 32 7C EC A9 08 F1 ED 5E :A3
4110 EC A9 08 F3 ED 5C EC 5C C3 FF FF ED 5C 83 00 00 :AE
4120 10 2C 00 41 EC 5E C3 FF FF ED 5E 83 00 00 10 2C :92
4130 00 06 4F 5F 32 C4 35 C0 CC FF FF ED 5C CC 00 50 :CE
4140 34 06 EC 5E 35 10 17 DF 18 34 06 30 A9 01 71 1F :7B
4150 10 E3 E1 34 06 EC 5C C3 00 01 ED 5C 35 10 E6 8B :19
4160 1D 10 26 FF D8 CC 00 50 34 06 EC 5E 35 10 17 DE :04
4170 F0 34 06 30 A9 01 71 1F 10 E3 E1 E3 5C 34 06 EC :CD
4180 44 34 06 17 01 45 32 64 83 00 00 10 27 00 13 EC :2A
4190 5E ED A9 08 F1 EC 5C ED A9 08 F3 CC 00 01 32 C4 :89
41A0 35 C0 16 FF 71 35 D6 34 40 33 E4 32 7E EC A9 01 :57
41B0 6B 83 00 01 ED 5E EC 5E A3 A9 01 67 10 25 00 22 :8F
41C0 EC 5E 34 06 EC 44 34 06 17 01 00 32 64 83 00 00 :1F
41D0 10 27 00 04 EC 5E 35 D0 EC 5E C3 FF FF ED 5E 16 :F6
41E0 FF D4 4F 5F 35 D0 34 40 33 E4 17 ED E3 EC 44 34 :5C
41F0 06 17 00 04 32 62 35 C0 34 40 33 E4 32 7C 4F 5F :91
-------------------------------------------------------------
Sum: C2 40 CD A6 8A 1F 1D C9 95 EC ED 4A 5C 7F 50 CA :B1
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4200 ED A9 08 F1 EC 44 83 00 01 1F 01 E6 84 1D 10 27 :21
4210 00 1C EC 44 34 06 17 F4 05 32 62 ED 5C 34 06 EC :99
4220 44 A3 E1 ED A9 08 F3 EC 5C ED 44 16 00 06 4F 5F :9C
4230 ED A9 08 F3 4F 5F ED 5E EC 5E 83 00 04 10 2C 00 :97
4240 34 EC 44 A3 A9 01 67 10 27 00 20 EC 44 A3 A9 01 :EC
4250 6D 10 27 00 16 EC 44 34 06 17 F3 C2 32 62 ED 44 :B5
4260 EC A9 08 F1 C3 00 01 ED A9 08 F1 EC 5E C3 00 01 :EF
4270 ED 5E 16 FF C3 EC 44 A3 A9 01 6B 10 22 00 25 EC :4E
4280 A9 01 6B A3 44 10 23 00 18 AE A9 01 6B E6 82 AF :21
4290 A9 01 6B 1D AE A9 01 6D E7 82 AF A9 01 6D 16 FF :3B
42A0 DE 16 00 22 EC A9 01 6D A3 44 10 24 00 18 AE A9 :A3
42B0 01 6D E6 80 AF A9 01 6D 1D AE A9 01 6B E7 80 AF :90
42C0 A9 01 6B 16 FF DE 17 EA 4E 35 D6 34 40 33 E4 E6 :D3
42D0 D8 04 1D 10 27 00 1D AE 46 E6 80 AF 46 1D 34 06 :F3
42E0 AE 44 E6 80 AF 44 1D A3 E1 10 27 00 04 4F 5F 35 :0A
42F0 C0 16 FF DB CC 00 01 35 C0 34 40 33 E4 32 7C 4F :FA
-------------------------------------------------------------
Sum: B8 F8 8F 8B 8B B7 E2 C9 C1 3D 67 78 1F 52 05 1A :24
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4300 5F ED A9 08 F3 17 01 09 ED 5E A3 A9 08 F1 A3 44 :88
4310 10 2F 00 11 EC 44 34 06 EC A9 01 67 34 06 17 00 :08
4320 45 32 64 35 D6 EC 5E A3 A9 08 F1 C3 00 18 A3 44 :37
4330 10 2E 00 1E EC 5E A3 A9 08 F1 C3 00 17 34 06 EC :EB
4340 44 A3 E1 34 06 EC A9 01 6D 34 06 17 00 18 32 64 :04
4350 35 D6 EC 44 A3 5E E3 A9 08 F1 ED A9 08 F1 4F 5F :FE
4360 ED A9 08 F3 35 D6 34 40 33 E4 EC 46 C3 FF FF ED :07
4370 46 83 00 00 10 2F 00 28 EC 44 A3 A9 01 69 10 26 :4C
4380 00 10 EC A9 01 69 34 06 17 F2 93 32 62 ED 44 16 :C0
4390 00 0E AE 44 E6 80 AF 44 1D 10 26 FF F5 16 FF CA :7F
43A0 4F 5F ED A9 08 F3 ED A9 08 F1 EC 44 34 06 17 FE :4D
43B0 35 32 62 35 C0 34 40 33 E4 CC 00 18 34 06 4F 5F :15
43C0 34 06 17 0E 04 32 64 EC A9 01 6D A3 A9 01 6B 34 :E8
43D0 06 17 00 3D 34 06 30 8C 02 20 2D 20 6C 69 6E 65 :67
43E0 20 25 64 20 20 25 64 20 62 79 74 65 20 66 72 65 :A3
43F0 65 2E 20 20 20 20 20 20 20 20 20 20 20 20 20 20 :53
-------------------------------------------------------------
Sum: B3 40 66 2D B6 81 1E 4B 6B C6 AD 57 33 B3 07 A5 :ED
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4400 20 20 20 20 20 20 20 00 34 10 17 DE E5 32 66 35 :CB
4410 C0 34 40 33 E4 32 7C 4F 5F ED 5E EC A9 01 67 ED :DC
4420 5C EC 5C A3 A9 01 6B 10 24 00 19 E6 D8 FC 1D 10 :90
4430 26 00 07 EC 5E C3 00 01 ED 5E EC 5C C3 00 01 ED :7F
4440 5C 16 FF DD EC A9 08 F1 E3 5E C3 00 01 32 C4 35 :0C
4450 C0 34 40 33 E4 CC 00 50 34 06 EC A9 08 F1 35 10 :74
4460 17 DB FE 34 06 30 A9 01 71 1F 10 E3 E1 AE A9 08 :C7
4470 F3 E6 8B 1D 10 27 00 12 17 02 C5 34 06 EC A9 08 :7F
4480 F3 E3 E1 ED A9 08 F3 16 00 14 EC A9 08 F1 C3 00 :C3
4490 01 ED A9 08 F1 4F 5F ED A9 08 F3 17 EB 8D CC 00 :2A
44A0 50 34 06 EC A9 08 F1 35 10 17 DB B5 34 06 30 A9 :17
44B0 01 71 1F 10 E3 E1 AE A9 08 F3 E6 8B 1D 83 00 20 :E8
44C0 10 26 00 0E EC A9 08 F3 C3 00 01 ED A9 08 F3 16 :3F
44D0 FF CC EC A9 08 F7 C3 FF FF ED A9 08 F7 83 00 00 :38
44E0 10 2E FF 71 35 C0 34 40 33 E4 CC 00 50 34 06 EC :70
44F0 A9 08 F1 35 10 17 DB 69 34 06 30 A9 01 71 1F 10 :F6
-------------------------------------------------------------
Sum: 95 E8 16 91 50 99 83 30 2D DD 44 6A 4E 23 0D 4F :45
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4500 E3 E1 AE A9 08 F3 E6 8B 1D 83 00 00 10 26 00 17 :74
4510 EC A9 08 F1 C3 00 01 ED A9 08 F1 4F 5F ED A9 08 :2D
4520 F3 17 EB 07 16 00 54 EC A9 08 F3 C3 00 01 ED A9 :50
4530 08 F3 CC 00 50 34 06 EC A9 08 F1 35 10 17 DB 21 :37
4540 34 06 30 A9 01 71 1F 10 E3 E1 AE A9 08 F3 E6 8B :3B
4550 1D 10 27 00 26 CC 00 50 34 06 EC A9 08 F1 35 10 :A3
4560 17 DA FE 34 06 30 A9 01 71 1F 10 E3 E1 AE A9 08 :C6
4570 F3 E6 8B 1D 83 00 20 10 26 FF AC CC 00 50 34 06 :5B
4580 EC A9 08 F1 35 10 17 DA D8 34 06 30 A9 01 71 1F :40
4590 10 E3 E1 AE A9 08 F3 E6 8B 1D 83 00 20 10 26 00 :8D
45A0 0E EC A9 08 F3 C3 00 01 ED A9 08 F3 16 FF CC EC :C0
45B0 A9 08 F7 C3 FF FF ED A9 08 F7 83 00 00 10 2E FF :BE
45C0 29 35 C0 34 40 33 E4 CC 00 50 34 06 EC A9 08 F1 :8D
45D0 35 10 17 DA 8C 34 06 30 A9 01 71 1F 10 E3 E1 AE :E8
45E0 A9 08 F3 E6 8B 1D 83 00 00 10 26 00 14 EC A9 08 :9C
45F0 F1 C3 00 01 ED A9 08 F1 4F 5F ED A9 08 F3 17 EA :84
-------------------------------------------------------------
Sum: D0 FA A0 FA F5 9B 95 18 16 51 F7 39 67 98 A3 2D :07
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4600 2A CC 00 50 34 06 EC A9 08 F1 35 10 17 DA 52 34 :CA
4610 06 30 A9 01 71 1F 10 E3 E1 34 06 EC A9 08 F3 C3 :D1
4620 00 01 ED A9 08 F3 35 10 E6 8B 1D 83 00 20 10 27 :3F
4630 FF CF 17 01 0B 83 00 01 34 06 EC A9 08 F3 E3 E1 :03
4640 ED A9 08 F3 EC A9 08 F7 C3 FF FF ED A9 08 F7 83 :FE
4650 00 00 10 2E FF 71 35 C0 34 40 33 E4 CC 00 50 34 :7E
4660 06 EC A9 08 F1 35 10 17 D9 F7 34 06 30 A9 01 71 :45
4670 1F 10 E3 E1 AE A9 08 F3 E6 8B 1D 83 00 00 10 26 :8C
4680 00 14 EC A9 08 F1 C3 00 01 ED A9 08 F1 4F 5F ED :90
4690 A9 08 F3 17 E9 95 CC 00 50 34 06 EC A9 08 F1 35 :52
46A0 10 17 D9 BD 34 06 30 A9 01 71 1F 10 E3 E1 34 06 :6F
46B0 EC A9 08 F3 C3 00 01 ED A9 08 F3 35 10 E6 8B 1D :B8
46C0 83 00 20 10 27 FF CF CC 00 50 34 06 EC A9 08 F1 :8C
46D0 35 10 17 D9 8C 34 06 30 A9 01 71 1F 10 E3 E1 34 :6D
46E0 06 EC A9 08 F3 C3 00 01 35 10 E6 8B 1D 10 27 00 :64
46F0 3B CC 00 50 34 06 EC A9 08 F1 35 10 17 D9 62 34 :EA
-------------------------------------------------------------
Sum: DF 15 F1 B6 04 1B 07 9A 9A 63 48 7B 2A 39 11 EB :7A
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4700 06 30 A9 01 71 1F 10 E3 E1 34 06 EC A9 08 F3 C3 :D1
4710 00 01 35 10 E6 8B 1D 83 00 20 10 27 00 0E EC A9 :51
4720 08 F3 C3 00 01 ED A9 08 F3 16 FF 9B EC A9 08 F7 :94
4730 C3 FF FF ED A9 08 F7 83 00 00 10 2E FF 1E 35 C0 :29
4740 34 40 33 E4 32 7E 4F 5F ED 5E CC 00 50 34 06 EC :76
4750 A9 08 F1 35 10 17 D9 09 34 06 30 A9 01 71 1F 10 :94
4760 E3 E1 AE A9 08 F3 E6 8B 1D 34 06 17 DF FE 32 62 :66
4770 83 00 00 10 27 00 74 CC 00 50 34 06 EC A9 08 F1 :12
4780 35 10 17 D8 DC 34 06 30 A9 01 71 1F 10 E3 E1 34 :BC
4790 06 EC 5E E3 A9 08 F3 35 10 E6 8B 1D 34 06 17 DF :DA
47A0 CB 32 62 83 00 00 10 26 00 33 CC 00 50 34 06 EC :8D
47B0 A9 08 F1 35 10 17 D8 A9 34 06 30 A9 01 71 1F 10 :33
47C0 E3 E1 34 06 EC 5E E3 A9 08 F3 35 10 E6 8B 1D 34 :D6
47D0 06 17 DF 48 32 62 83 00 00 10 27 00 0A EC 5E C3 :A9
47E0 00 01 ED 5E 16 FF 90 EC 5E 35 D0 CC 00 50 34 06 :96
47F0 EC A9 08 F1 35 10 17 D8 68 34 06 30 A9 01 71 1F :CE
-------------------------------------------------------------
Sum: 98 24 42 E0 70 49 3D 51 CD DE 85 93 DE 7F B8 9D :9A
```

リスト 2　vimマシン語ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4800 10 E3 E1 AE A9 08 F3 E6 8B 1D 34 06 17 DF 0D 32 :23
4810 62 83 00 00 10 27 00 41 CC 00 50 34 06 EC A9 08 :50
4820 F1 35 10 17 D8 3B 34 06 30 A9 01 71 1F 10 E3 E1 :D8
4830 34 06 EC 5E E3 A9 08 F3 35 10 E6 8B 1D 34 06 17 :2F
4840 DE DA 32 62 83 00 00 10 27 00 0A EC 5E C3 00 01 :1E
4850 ED 5E 16 FF C3 EC 5E 35 D0 CC 00 01 35 D0 34 40 :B8
4860 33 E4 32 7F EC A9 08 F3 83 00 00 10 2D 00 31 CC :15
4870 00 50 34 06 EC A9 08 F1 35 10 17 D7 E4 34 06 30 :99
4880 A9 01 71 1F 10 E3 E1 34 06 EC A9 08 F3 C3 FF FF :99
4890 ED A9 08 F3 35 10 E6 8B 1D 83 00 20 10 27 FF C4 :01
48A0 EC A9 08 F3 83 00 00 10 2C 00 15 EC A9 08 F1 C3 :B5
48B0 FF FF ED A9 08 F1 CC 00 4F ED A9 08 F3 17 E7 6B :A2
48C0 CC 00 50 34 06 EC A9 08 F1 35 10 17 D7 93 34 06 :E4
48D0 30 A9 01 71 1F 10 E3 E1 AE A9 08 F3 E6 8B 1D E7 :05
48E0 5F 34 06 17 DE 86 32 62 83 00 00 10 26 00 11 E6 :58
48F0 5F 1D 34 06 17 DE 25 32 62 83 00 00 10 27 00 60 :7E
-----------------------------------------------------------
Sum: D0 59 84 79 7C 95 13 95 8D 6F 0B 40 8F 24 42 93 :AE
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4900 EC A9 08 F3 83 00 00 10 2F 00 55 CC 00 50 34 06 :FD
4910 EC A9 08 F1 35 10 17 D7 48 34 06 30 A9 01 71 1F :AD
4920 10 E3 E1 34 06 EC A9 08 F3 83 00 01 35 10 E6 8B :D8
4930 1D E7 5F 34 06 17 DE 34 32 62 83 00 00 10 26 00 :13
4940 11 E6 5F 1D 34 06 17 DD D3 32 62 83 00 00 10 27 :C2
4950 00 0E EC A9 08 F3 C3 FF FF ED A9 08 F3 16 FF A0 :A5
4960 EC A9 08 F7 C3 FF FF ED A9 08 F7 83 00 00 10 2E :AB
4970 FE F2 35 C2 34 40 33 E4 32 7F EC A9 08 F3 83 00 :36
4980 00 10 2D 00 31 CC 00 50 34 06 EC A9 08 F1 35 10 :97
4990 17 D6 CE 34 06 30 A9 01 71 1F 10 E3 E1 34 06 EC :59
49A0 A9 08 F3 C3 FF FF ED A9 08 F3 35 10 E6 8B 1D 83 :4C
49B0 00 20 10 27 FF C4 EC A9 08 F3 83 00 00 10 2C 00 :69
49C0 15 EC A9 08 F1 C3 FF FF ED A9 08 F1 CC 00 4F ED :FB
49D0 A9 08 F3 17 E6 55 EC A9 08 F3 83 00 00 10 2F 00 :48
49E0 3B CC 00 50 34 06 EC A9 08 F1 35 10 17 D6 72 34 :F7
49F0 06 30 A9 01 71 1F 10 E3 E1 34 06 EC A9 08 F3 83 :91
-----------------------------------------------------------
Sum: BF A9 1B 59 A8 47 13 A7 DC 8B 46 3D 34 28 BA C8 :4D
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4A00 00 01 35 10 E6 8B 1D 83 00 20 10 27 00 0E EC A9 :51
4A10 08 F3 C3 FF FF ED A9 08 F3 16 FF BA EC A9 08 F7 :B0
4A20 C3 FF FF ED A9 08 F7 83 00 00 10 2E FF 4C 35 C2 :59
4A30 34 40 33 E4 32 7E 17 D8 96 83 00 3E 10 27 00 02 :BA
4A40 35 C6 4F 5F ED A9 08 F3 4F 5F ED 5E EC 5E A3 A9 :C9
4A50 08 F9 10 2C 00 19 CC 00 01 34 06 CC 00 20 34 06 :83
4A60 17 EC A5 32 64 EC 5E C3 00 01 ED 5E 16 FF DD EC :75
4A70 A9 08 F1 34 06 17 08 3E 32 62 EC A9 08 F7 C3 FF :23
4A80 FF ED A9 08 F7 83 00 00 10 2E 00 03 16 00 11 EC :6B
4A90 A9 08 F1 C3 00 01 ED A9 08 F1 17 E5 8E 16 FF A2 :36
4AA0 17 E3 5A 35 C6 34 40 33 E4 17 D8 23 83 00 3C 10 :BB
4AB0 27 00 02 35 C0 4F 5F ED A9 08 F3 EC A9 08 F9 34 :27
4AC0 06 17 EA 36 32 62 EC A9 08 F1 34 06 17 07 E7 32 :D0
4AD0 62 EC A9 08 F7 C3 FF FF ED A9 08 F7 83 00 00 10 :DF
4AE0 2E 00 03 16 00 11 EC A9 08 F1 C3 00 01 ED A9 08 :48
4AF0 F1 17 E5 37 16 FF BE 17 E3 03 35 C0 34 40 33 E4 :74
-----------------------------------------------------------
Sum: 69 D8 90 91 D3 FF 2F 0B 90 7B 01 32 A4 F0 A8 FE :E6
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4B00 32 7C CC 00 18 34 06 4F 5F 34 06 17 06 BB 32 64 :22
4B10 CC 00 3A 34 06 17 D7 C6 32 62 CC 00 18 34 06 CC :72
4B20 00 01 34 06 17 E3 62 32 64 4F 5F ED 5C ED 5E 30 :9F
4B30 A9 08 FB 1F 10 ED A9 01 6F 17 05 5B ED 5C E6 B9 :40
4B40 01 6F 1D 83 00 2C 10 27 00 03 16 00 16 EC 5C ED :D7
4B50 5E 4F 5F ED 5C EC A9 01 6F C3 00 01 ED A9 01 6F :24
4B60 16 FF D6 AE A9 01 6F E6 80 AF A9 01 6F 1D 10 83 :90
4B70 00 71 10 26 00 1D E6 B9 01 6F 1D 10 27 00 03 16 :40
4B80 00 F0 CC 6F FC 1F 01 4F 5F E7 84 17 D6 24 17 D4 :5C
4B90 8F 20 08 10 83 00 77 10 26 00 27 E6 B9 01 6F 1D :4A
4BA0 10 27 00 0E EC A9 01 6F 34 06 17 E4 1D 32 62 16 :46
4BB0 00 0D AE A9 01 65 EC 02 34 06 17 E4 0D 32 62 16 :A4
4BC0 00 B0 10 83 00 74 10 26 00 09 EC 5C ED A9 08 F9 :D5
4BD0 16 00 9F 10 83 00 00 10 26 00 12 EC 5C 10 27 00 :0F
4BE0 09 EC 5C 34 06 17 F7 11 32 62 16 00 85 10 83 00 :6C
4BF0 6D 10 26 00 14 EC 5C 34 06 EC 5E 34 06 4F 5F 34 :9F
-----------------------------------------------------------
Sum: 47 A3 4A 9A 53 F5 BE 5A 9F 2A 5D B2 8D 8B 47 58 :BD
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4C00 06 17 00 70 32 66 16 00 69 10 83 00 63 10 26 00 :D0
4C10 27 AE A9 01 6F E6 80 AF A9 01 6F 1D 83 00 6F 10 :3B
4C20 26 00 12 EC 5C 34 06 EC 5E 34 06 CC 00 01 34 06 :45
4C30 17 00 41 32 66 16 00 3A 10 83 00 73 10 26 00 10 :8C
4C40 EC 5C 34 06 EC 5E 34 06 17 02 9E 32 64 16 00 22 :8B
4C50 10 83 00 64 10 26 00 10 EC 5C 34 06 EC 5E 34 06 :43
4C60 17 02 2F 32 64 16 00 0A CC 00 07 34 06 17 D6 6E :66
4C70 32 62 35 D6 34 40 33 E4 32 72 EC 48 83 00 00 10 :95
4C80 26 00 07 17 01 DD 32 C4 35 C0 EC 46 83 00 00 10 :D2
4C90 26 00 04 EC 48 ED 46 17 03 FD ED 5E EC 46 A3 48 :10
4CA0 10 2E 00 10 EC 5E A3 46 10 2D 00 3C EC 5E A3 48 :2F
4CB0 10 2C 00 34 CC 00 18 34 06 4F 5F 34 06 17 05 09 :9B
4CC0 32 64 30 8C 02 20 16 07 49 6C 6C 65 67 61 6C 20 :6B
4CD0 6C 69 6E 65 20 6E 75 6D 62 65 72 2E 00 34 10 17 :DA
4CE0 D6 10 32 62 32 C4 35 C0 17 01 4B EC 5E C3 00 01 :D6
4CF0 34 06 17 01 0B 32 62 ED 54 EC 46 34 06 17 01 00 :B6
-----------------------------------------------------------
Sum: C3 45 86 9C 57 1C 58 4F E5 8F 64 D7 FB EC 9B AD :22
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4D00 32 62 ED 58 34 06 EC 48 C3 00 01 34 06 17 00 F0 :4C
4D10 32 62 ED 56 A3 E1 ED 5A EC 5A 34 06 EC 54 34 06 :9C
4D20 17 00 6A 32 64 EC 5E A3 46 10 2C 00 06 EC 58 E3 :B3
4D30 5A ED 58 4F 5F ED 5C EC 5C A3 5A 10 2C 00 1D EC :20
4D40 5C AE 58 E6 8B 1D 34 06 EC 5C AE 54 30 8B 35 06 :6A
4D50 E7 84 EC 5C C3 00 01 ED 5C 16 FF DB EC 44 83 00 :63
4D60 00 10 26 00 19 EC 5A 34 06 EC 58 34 06 17 00 5B :BF
4D70 32 64 EC 54 34 06 17 F4 7F 32 62 16 00 0B EC 5A :95
4D80 E3 54 34 06 17 F4 71 32 62 32 C4 35 C0 34 40 33 :13
4D90 E4 32 7E EC A9 01 6B 83 00 01 ED 5E EC 5E A3 44 :95
4DA0 10 25 00 1B AE 5E E6 84 1D 34 06 EC 46 AE 5E 30 :8B
4DB0 8B 35 06 E7 84 EC 5E C3 FF FF ED 5E 16 FF DD EC :65
4DC0 A9 01 6B E3 46 ED A9 01 6B 35 C6 34 40 33 E4 32 :F8
4DD0 7E EC A9 01 6B A3 46 ED A9 01 6B EC 44 ED 5E EC :D1
4DE0 5E A3 A9 01 6B 10 24 00 15 EC 46 AE 5E E6 8B 1D :2B
4DF0 AE 5E E7 84 EC 5E C3 00 01 ED 5E 16 FF E1 35 C6 :C1
-----------------------------------------------------------
Sum: DF 25 4E 22 2F 0C 2F 36 C6 12 9B 84 2F 6E 6D 14 :29
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4E00 34 40 33 E4 32 7E EC A9 01 67 ED 5E EC 5E A3 A9 :19
4E10 01 6B 10 24 00 1C EC 44 C3 FF FF ED 44 83 00 00 :61
4E20 10 2F 00 0E AE 5E E6 80 AF 5E 1D 10 26 FF F5 16 :29
4E30 FF DA EC 5E 35 D0 34 40 33 E4 17 E1 93 EC A9 01 :D4
4E40 6D A3 A9 01 69 10 24 00 18 AE A9 01 6D E6 80 AF :49
4E50 A9 01 6D 1D AE A9 01 6B E7 80 AF A9 01 6B 16 FF :37
4E60 DC 35 C0 34 40 33 E4 CC 00 18 34 06 4F 5F 34 06 :62
4E70 17 03 56 32 64 30 8C 02 20 0F 07 53 79 6E 74 61 :09
4E80 78 20 65 72 72 6F 72 2E 00 34 10 17 D4 64 32 62 :17
4E90 35 C0 34 40 33 E4 32 7C EC 46 83 00 00 10 26 00 :19
4EA0 05 17 FF BF 35 D6 EC 44 83 00 00 10 26 00 04 EC :BE
4EB0 46 ED 44 17 FF 80 EC 44 34 06 17 FF 43 32 62 ED :51
4EC0 5E EC 46 C3 00 01 34 06 17 FF 35 32 62 ED 5C EC :A2
4ED0 5C A3 5E 34 06 EC 5E 34 06 17 FE EF 32 64 EC 5E :FF
4EE0 34 06 17 F3 13 32 62 35 D6 34 40 33 E4 32 E8 E9 :84
4EF0 EC 46 83 00 00 10 26 00 07 17 FF 67 32 C4 35 C0 :5A
-----------------------------------------------------------
Sum: 1F 4F 75 6A C2 BC 1D 87 62 DE CF 20 06 D7 A2 03 :20
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4F00 EC 44 83 00 00 10 26 00 04 EC 46 ED 44 4F 5F ED :EB
4F10 56 ED 58 ED 5A AE A9 01 6F E6 80 AF A9 01 6F 1D :F4
4F20 E7 55 EC A9 01 6F ED C8 EF E6 B9 01 6F 1D 10 27 :48
4F30 00 25 E6 55 1D 34 06 E6 B9 01 6F 1D A3 E1 10 27 :9E
4F40 00 15 EC A9 01 6F C3 00 01 ED A9 01 6F EC 5A C3 :ED
4F50 00 01 ED 5A 16 FF D2 E6 B9 01 6F 1D 83 00 00 10 :EE
4F60 26 00 07 17 FE FD 32 C4 35 C0 4F 5F AE A9 01 6F :9F
4F70 E7 80 AF A9 01 6F EC A9 01 6F ED C8 ED E6 B9 01 :76
4F80 6F 1D 10 27 00 25 E6 55 1D 34 06 E6 B9 01 6F 1D :A6
4F90 A3 E1 10 27 00 15 EC A9 01 6F C3 00 01 ED A9 01 :30
4FA0 6F EC 58 C3 00 01 ED 58 16 FF D2 E6 B9 01 6F 1D :CF
4FB0 10 27 00 0C 4F 5F AE A9 01 6F E7 80 AF A9 01 6F :E7
4FC0 E6 B9 01 6F 1D 83 00 67 10 26 00 05 CC 00 01 ED :0B
4FD0 56 17 FE 62 EC 44 34 06 17 FE 25 32 62 ED 53 ED :32
4FE0 C8 EB EC 46 C3 00 01 34 06 17 FE 14 32 62 ED 51 :DE
4FF0 ED C8 E9 EC C8 EB A3 51 10 24 00 8D EC C8 EB 34 :C5
-----------------------------------------------------------
Sum: B8 D5 88 CE 71 87 BA F3 7D 46 E7 23 FA 78 B6 A4 :21
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5000 06 EC C8 EF 34 06 17 F2 C2 32 64 83 00 00 10 27 :FE
5010 00 6B EC C8 EB ED C8 E9 EC 5A A3 58 10 2F 00 10 :38
5020 EC 5A A3 58 34 06 EC C8 EB 34 06 17 FD 9D 32 64 :9B
5030 EC 5A A3 58 10 2C 00 10 EC 58 A3 5A 34 06 EC C8 :BC
5040 EB 34 06 17 FD 47 32 64 4F 5F ED 5C EC 5C A3 58 :50
5050 10 2C 00 1A EC 5C AE C8 ED E6 8B 1D AE C8 EB E7 :D7
5060 80 AF C8 EB EC 5C C3 00 01 ED 5C 16 FF DE EC 56 :6C
5070 83 00 00 10 26 00 03 16 00 0F 16 00 09 EC C8 EB :9F
5080 C3 00 01 ED C8 EB 16 FF 6A EC C8 E9 34 06 17 F1 :C2
5090 67 32 62 32 C4 35 C0 34 40 33 E4 32 7E 17 00 50 :88
50A0 ED 5E E6 B9 01 6F 1D 83 00 2B 10 26 00 19 EC A9 :09
50B0 01 6F C3 00 01 ED A9 01 6F 17 00 34 34 06 EC 5E :09
50C0 E3 E1 ED 5E 16 00 25 E6 B9 01 6F 1D 83 00 2D 10 :36
50D0 26 00 19 EC A9 01 6F C3 00 01 ED A9 01 6F 17 00 :25
50E0 0F 34 06 EC 5E A3 E1 ED 5E 16 00 00 EC 5E 35 D0 :C7
50F0 34 40 33 E4 32 7E E6 B9 01 6F 1D 83 00 2E 10 26 :4E
-----------------------------------------------------------
Sum: 40 6E 13 85 3B C2 68 FB F3 41 CF 99 39 F7 E8 31 :8B
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5100 00 10 EC A9 01 6F C3 00 01 ED A9 01 6F 17 F3 01 :EA
5110 35 D0 E6 B9 01 6F 1D 83 00 24 10 26 00 10 EC A9 :B3
5120 01 6F C3 00 01 ED A9 01 6F 17 00 40 35 D0 4F 5F :44
5130 ED 5E E6 B9 01 6F 1D 34 06 17 D5 E0 32 62 83 00 :94
5140 00 10 27 00 23 AE A9 01 6F E6 80 AF A9 01 6F 1D :6C
5150 34 06 EC 5E 58 49 34 06 58 49 58 49 E3 E1 E3 E1 :29
5160 83 00 30 ED 5E 16 FF CA EC 5E 35 D0 34 40 33 E4 :B7
5170 32 7C 4F 5F ED 5C EC A9 01 67 ED 5E EC 5E A3 A9 :B3
5180 01 6B 10 24 00 15 AE 5E E6 80 AF 5E 1D 10 26 FF :86
5190 F5 EC 5C C3 00 01 ED 5C 16 FF E1 EC A9 01 6D ED :30
51A0 5E EC 5E A3 A9 01 69 10 24 00 15 AE 5E E6 80 AF :C8
51B0 5E 1D 10 26 FF F5 EC 5C C3 00 01 ED 5C 16 FF E1 :F0
51C0 EC 5C C3 00 18 32 C4 35 C0 34 40 33 E4 32 7E CC :15
51D0 FC 82 ED 5E 17 CF C0 CC 00 03 AE 5E E7 80 AF 5E :BE
51E0 CC 00 05 AE 5E E7 80 AF 5E CC 00 12 AE 5E E7 80 :A2
51F0 AF 5E EC 44 AE 5E E7 80 AF 5E EC 46 AE 5E E7 80 :62
-----------------------------------------------------------
Sum: 21 DB 88 C5 AD F5 49 88 DA 13 08 3B 29 54 E6 3A :89
```

リスト 2　vimマシン語ダンプ・リスト

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5200 AF 5E CC 00 11 AE 5E E7 80 AF 5E EC A9 08 F5 C3 :BF
5210 00 80 AE 5E E7 80 AF 5E 17 CF 8C 35 C6 34 40 33 :14
5220 E4 32 7C CC FC 82 ED 5E 17 CF 6C 30 8D 00 33 1F :88
5230 10 ED 5C 30 8D 00 4E 1F 10 34 06 EC 5C A3 E1 10 :A9
5240 24 00 14 E6 D8 FC 1D AE 5E E7 80 AF 5E EC 5C C3 :9A
5250 00 01 ED 5C 16 FF DC 17 CF 4D 17 CF 3A 17 CF 47 :BB
5260 35 D6 3F 59 41 4D 41 55 43 48 49 93 D3 8F 90 F7 :17
5270 D4 0A DC 1F 93 3B B5 D4 09 DD 1F FD D4 0E B7 D4 :9F
5280 09 B6 D4 0A 39 34 40 33 E4 E6 45 1D 83 00 20 10 :5C
5290 2C 00 0A E6 45 1D 83 00 00 10 2C 00 0D E6 45 1D :92
52A0 34 06 17 D0 39 32 62 16 00 0A CC 00 20 34 06 17 :4B
52B0 D0 2C 32 62 35 C0 34 40 33 E4 32 79 CC 00 50 34 :0B
52C0 06 EC 44 35 10 17 CD 99 34 06 30 A9 01 71 1F 10 :AC
52D0 E3 E1 ED 5B CC FC 82 ED 5D 17 CE BB CC 00 07 AE :C1
52E0 5D E7 80 AF 5D 4F 5F AE 5D E7 80 AF 5D EC 44 AE :DA
52F0 5D E7 80 AF 5D CC 00 4F AE 5D E7 80 AF 5D EC 44 :99
---------------------------------------------------------
Sum: AC 61 C6 24 C5 A4 3E BC EA 1F 2F 74 EC 53 CC 22 :33
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5300 AE 5D E7 80 AF 5D CC 00 07 AE 5D E7 80 AF 5D CC :9B
5310 00 50 AE 5D E7 80 AF 5D 4F 5F ED 59 EC 59 83 00 :8A
5320 50 10 2C 00 39 EC 59 AE 5B E6 8B 1D E7 5F 83 00 :6A
5330 20 10 2C 00 16 E6 5F 1D 83 00 00 10 2D 00 0C CC :6C
5340 00 20 AE 5D E7 80 AF 5D 16 00 09 E6 5F 1D AE 5D :2A
5350 E7 80 AF 5D EC 59 C3 00 01 ED 59 16 FF BE 17 CE :7A
5360 46 32 C4 35 C0 34 40 33 E4 32 7C CC FC 82 ED 5E :FF
5370 17 CE 24 CC 00 07 AE 5E E7 80 AF 5E 4F 5F AE 5E :16
5380 E7 80 AF 5E EC 44 AE 5E E7 80 AF 5E CC 00 4F AE :ED
5390 5E E7 80 AF 5E EC 44 AE 5E E7 80 AF 5E CC 00 07 :55
53A0 AE 5E E7 80 AF 5E CC 00 50 AE 5E E7 80 AF 5E 4F :6B
53B0 5F ED 5C EC 5C 83 00 50 10 2C 00 13 CC 00 20 AE :AC
53C0 5E E7 80 AF 5E EC 5C C3 00 01 ED 5C 16 FF E4 17 :37
53D0 CD D5 35 D6 39 00 00 00 00 00 00 00 00 00 00 00 :E6
53E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
53F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: DF DB 59 96 64 C0 AD 35 BB D4 DC F6 B5 9D 80 48 :2A
```

リスト 3　vil. Oマシン語ダンプ・リスト

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1400 34 7D 86 00 1F 8B 86 01 97 BF BD D0 72 35 FD FB :EA
1410 FA 25 06 6D 07 27 F6 A6 06 35 FD 34 7F C6 00 1F :2C
1420 9B C6 02 D7 BF BD D0 8E 35 FF CC 00 01 ED 04 AD :B3
1430 9F FB FA 35 FF 34 7F 86 00 1F 8B BD D8 07 25 0C :78
1440 A6 80 81 00 26 FA 86 0D A7 82 35 FF 7E 94 8D 00 :56
1450 17 02 8D 30 8D 02 69 A6 03 E6 88 16 C0 11 A2 88 :F6
1460 1A 34 06 43 25 01 30 8D 02 8E 10 32 62 35 90 12 :85
1470 F5 FD 0F BD D0 8E B7 FD 0F 39 00 00 00 00 00 00 :18
1480 16 FF 98 16 FF 7A 16 FF AC 39 0C 35 10 20 02 8D :36
1490 EF A6 80 26 FA 6E 84 86 0D 8D E5 86 0A 20 E1 36 :F3
14A0 06 A6 43 3D 34 06 EC 41 3D EB E4 E7 E4 A6 C4 E6 :BA
14B0 43 3D EB E0 A6 E0 1E 89 33 44 39 8E 00 00 34 10 :FA
14C0 34 10 34 10 4D 2A 04 8D 40 63 61 6C 60 2B 40 58 :23
14D0 49 2A F8 44 56 ED 64 EC C1 2A 04 8D 2C 63 61 6A :18
14E0 60 2B 1E 58 49 2A F8 44 56 A3 64 20 06 58 49 24 :F8
14F0 F8 E3 64 1C FE 2B 02 1A 01 69 63 69 62 6A 60 2A :2C
---------------------------------------------------------
Sum: 57 E6 9F CA 49 68 A7 1E 0E CF 18 BA 5C FF 0A 36 :66
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1500 EC EC 62 6D 61 32 66 2A 05 43 53 C3 00 01 39 32 :94
1510 66 CC 7F FF 39 4D 2B F1 39 A3 C1 2F 2C CC 00 00 :16
1520 39 A3 C1 2E 24 CC 00 00 39 A3 C1 2D 1C CC 00 00 :6D
1530 39 A3 C1 2C 14 CC 00 00 39 A3 C1 27 0C CC 00 00 :45
1540 39 A3 C1 26 04 CC 00 00 39 CC 00 01 39 83 00 00 :55
1550 27 F7 CC 00 00 39 17 FF 2A 1F 89 4F 39 8D 15 16 :4B
1560 FF A7 17 FE EB 17 FF 1E A6 80 81 24 27 2B 81 2D :A5
1570 27 EB 30 1F CC 00 00 34 06 A6 84 82 30 2B 18 81 :07
1580 09 22 14 A7 84 EC 60 58 49 58 49 E3 60 58 49 EB :C7
1590 80 89 00 ED 60 20 E2 35 86 CC 00 00 34 06 A6 84 :43
15A0 82 30 2B F3 81 09 23 06 82 07 81 0F 22 E9 A7 84 :D2
15B0 EC 60 58 49 58 49 58 49 58 49 EB 80 ED 60 20 DE :86
15C0 8E 00 00 36 10 36 10 36 10 36 10 4D 2A 05 63 40 :C5
15D0 17 00 D7 58 49 ED 46 8E 00 0F C6 05 1C FE A6 C5 :AF
15E0 49 81 09 22 04 1C FE 20 04 82 0A 1A 01 A7 C5 5A :A4
15F0 26 EC 68 47 69 46 24 02 6C 45 30 1F 26 DC C6 01 :5F
---------------------------------------------------------
Sum: 55 D2 16 D0 10 16 DC 2E E8 BD E9 39 2D F8 31 27 :81
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1600 A6 C5 26 05 5C C1 05 26 F7 39 8D B4 34 04 EB 49 :BB
1610 C2 06 23 02 8D 20 35 04 6D 40 2A 05 86 2D 17 FE :77
1620 5F A6 C5 8B 30 17 FE 58 5C C1 06 26 F4 33 4A 39 :E5
1630 33 5E 8D 8C 20 E2 86 20 17 FE 45 5A 26 FA 39 34 :93
1640 04 8D 04 35 04 1F 98 1F 89 44 44 44 44 8D 04 1F :ED
1650 98 84 0F 8B 30 81 39 23 02 8B 07 16 FE 22 1F 98 :44
1660 20 F9 33 5C A6 45 E6 47 3D ED 42 EC 45 3D EB 42 :C7
1670 89 00 ED 41 A6 44 E6 47 3D E3 41 ED 41 A6 44 E6 :2D
1680 46 3D EB 41 89 00 ED C4 39 34 06 EC 8C 1A 8E 3D :B9
1690 09 36 16 8D CD EC 42 33 48 C3 00 03 ED 8C 09 35 :D5
16A0 10 36 16 8D BD 33 48 39 1C 2C 6A 49 16 FE 5A 20 :E3
16B0 7C 4F 34 3B 8C 00 00 00 00 00 00 00 00 00 00 00 :C6
16C0 43 6F 70 79 72 69 67 68 74 20 28 63 29 20 31 39 :17
16D0 38 32 20 62 79 20 4E 2E 54 73 75 64 61 00 8E 10 :A0
16E0 30 8C DD C6 1D 4F AB 80 5A 26 FB 81 19 26 01 39 :6B
16F0 8E 80 00 4F A7 82 2D FC 2E FA BD 1F 34 35 76 12 :A4
---------------------------------------------------------
Sum: 53 7E 86 01 07 7C 5F B4 C9 AD 95 0B 02 0F FE B9 :CC
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1700 34 7F 86 6F 1F 8B 5F 1F 03 10 DF FE CC 20 00 DD :89
1710 00 CC 6F FC DD 02 17 FE 3D DD 04 DC 04 36 06 CC :31
1720 00 00 17 FE 1C 36 06 DC 00 36 06 CC 60 00 17 FD :C5
1730 F0 A4 C0 E4 C0 10 27 00 8B DC 04 36 06 CC 00 27 :C9
1740 17 FD FE 36 06 DC 04 36 06 CC 00 00 17 FD F2 A4 :E0
1750 C0 E4 C0 36 06 DC 00 36 06 CC 60 00 17 FD C2 A4 :5E
1760 C0 E4 C0 10 27 00 08 17 FD EC DD 04 16 FF CA DC :3F
1770 04 36 06 CC 00 00 17 FD C0 36 06 DC 00 36 06 CC :00
1780 60 00 17 FD 94 AA C0 EA C0 10 27 00 03 16 00 34 :A0
1790 17 FD C3 DD 04 DC 04 83 00 0D 10 27 00 15 DC 04 :54
17A0 9E 00 E7 84 DC 00 C3 00 01 DD 00 17 FD A8 DD 04 :23
17B0 16 FF E2 CC 00 00 9E 00 E7 84 DC 00 C3 00 01 DD :49
17C0 00 16 FF 57 CC 20 00 DD 06 CC 80 00 DD 08 CC 00 :38
17D0 00 F7 FD 0F DC 06 93 00 10 2C 00 1A 9E 06 4F E6 :A7
17E0 84 9E 08 E7 84 DC 08 C3 00 01 DD 08 DC 06 C3 00 :C7
17F0 01 DD 06 16 FF DE DC 08 9E 02 ED 84 4F F6 FD 0F :1D
---------------------------------------------------------
Sum: 6F 6E FD 22 AA F1 62 8E F0 32 8D A0 E3 2E 36 CB :E8
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1800 DD 0A DC 04 83 00 00 10 26 00 05 10 DE FE 35 FF :A5
1810 17 FD 43 DD 04 DC 04 36 06 CC 00 00 17 FD 22 36 :8C
1820 06 DC 00 36 06 CC 60 00 17 FC F6 A4 C0 E4 C0 10 :6B
1830 27 00 8B DC 04 36 06 CC 00 27 17 FD 04 36 06 DC :F1
1840 04 36 06 CC 00 00 17 FC F8 A4 C0 E4 C0 36 06 DC :37
1850 00 36 06 CC 60 00 17 FC C8 A4 C0 E4 C0 10 27 00 :82
1860 08 17 FC F2 DD 04 16 FF CA DC 04 36 06 CC 00 00 :B5
1870 17 FC C6 36 06 DC 00 36 06 CC 60 00 17 FC 9A AA :B0
1880 C0 EA C0 10 27 00 03 16 00 34 17 FC C9 DD 04 DC :87
1890 04 83 00 0D 10 27 00 15 DC 04 9E 00 E7 84 DC 00 :A5
18A0 C3 00 01 DD 00 17 FC AE DD 04 16 FF E2 CC 00 00 :06
18B0 9E 00 E7 84 DC 00 C3 00 01 DD 00 16 FF 57 CC 20 :DE
18C0 00 DD 06 CC C0 00 DD 08 CC 00 00 F7 FD 0F DC 06 :05
18D0 93 00 10 2C 00 1A 9E 06 4F E6 84 9E 08 E7 84 DC :33
18E0 08 C3 00 01 DD 08 DC 06 C3 00 01 DD 06 16 FF DE :2D
18F0 DC 08 9E 02 ED 84 4F F6 FD 0F DD 0A 10 DE FE 35 :4E
---------------------------------------------------------
Sum: E0 77 D4 2C 71 A2 16 22 68 ED 23 3C 02 91 ED 98 :6E
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1900 FF 10 DE FE 35 FF 27 00 00 00 00 00 00 00 00 00 :46
---------------------------------------------------------
Sum: FF 10 DE FE 35 FF 27 00 00 00 00 00 00 00 00 00 :46
```

リスト 4　vis. Oマシン語ダンプ・リスト（$1400～$16FFは、リスト 3 と同じです）

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1700 34 7F 86 6F 1F 8B 5F 1F 03 10 DF FE CC 00 0D 17 :B0
1710 FF 4C CC 00 0A 17 FF 46 CC 6F FC DD 00 CC 00 00 :5D
1720 DD 02 CC 80 00 DD 04 CC 20 00 DD 06 9E 00 EC 84 :E9
1730 DD 08 CC 00 00 F7 FD 0F DC 04 36 06 DC 08 17 FD :C8
1740 E0 36 06 DC 06 36 06 CC 60 00 17 FD D4 A4 C0 E4 :96
1750 C0 10 27 00 1A 9E 04 4F E6 84 9E 06 E7 84 DC 06 :5D
1760 C3 00 01 DD 06 DC 04 C3 00 01 DD 04 16 FF C9 4F :59
1770 F6 FD 0F DD 0A CC 20 00 DD 0C DC 0C 93 06 10 2C :7B
1780 00 60 DC 02 C3 00 01 DD 02 17 FE A4 17 FC FC 20 :C9
1790 27 00 9E 0C 4F E6 84 36 06 CC 00 00 17 FD A2 36 :7E
17A0 06 DC 0C 36 06 DC 06 17 FD 77 A4 C0 E4 C0 10 27 :D6
17B0 00 12 9E 0C 4F E6 84 17 FE A4 DC 0C C3 00 01 DD :B7
17C0 0C 16 FF CE DC 0C 93 06 10 2C 00 13 DC 0C C3 00 :6A
17D0 01 DD 0C CC 00 0D 17 FE 85 CC 00 0A 17 FE 7F 16 :DD
17E0 FF 98 DC 04 93 08 10 2C 00 84 CC 20 00 DD 06 CC :6D
17F0 00 00 F7 FD 0F DC 04 93 08 10 2C 00 1A 9E 04 4F :C5
```

```
---------------------------------------------------------
Sum: 7F F1 29 70 3E 97 5A 22 8E 9E D2 A7 8C 3F 80 88 :D2
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1800 E6 84 9E 06 E7 84 DC 06 C3 00 01 DD 06 DC 04 C3 :A5
1810 00 01 DD 04 16 FF DE 4F F6 FD 0F DD 0A CC 20 00 :F9
1820 DD 0C DC 0C 93 06 10 2C 00 44 DC 02 C3 00 01 DD :69
1830 02 17 FD FC 17 FC 54 20 27 00 9E 0C 4F E6 84 83 :A6
1840 00 00 10 27 00 12 9E 0C 4F E6 84 17 FE 10 DC 0C :B9
1850 C3 00 01 DD 0C 16 FF E2 DC 0C C3 00 01 DD 0C CC :05
1860 00 0D 17 FD F9 CC 00 0A 17 FD F3 16 FF B4 CC 00 :8C
1870 00 17 FD EA 10 DE FE 35 FF 20 00 00 00 00 00 00 :3E
---------------------------------------------------------
Sum: 88 CC 79 FD BC 57 B9 CE 21 50 C4 F5 20 2F 5D FB :35
```

リスト5　vil2.Oマシン語ダンプ・リスト（$1400～$16FFは，リスト3と同じです）

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1700 34 7F 86 6F 1F 8B 5F 1F 03 10 DF FE CC 20 00 DD :89
1710 00 CC 6F FC DD 02 17 FE 3D DD 04 17 FE 38 DD 04 :77
1720 17 FE 33 DD 04 DC 04 36 06 CC 00 00 17 FE 12 36 :6E
1730 06 DC 00 36 06 CC 60 00 17 FD E6 A4 C0 E4 C0 10 :5C
1740 27 00 39 DC 04 83 00 0D 10 27 00 15 DC 04 9E 00 :9A
1750 E7 84 DC 00 C3 00 01 DD 00 17 FD FA DD 04 16 FF :EC
1760 E2 CC 00 00 9E 00 E7 84 DC 00 C3 00 01 DD 00 17 :4B
1770 FD E4 DD 04 17 FD DF DD 04 16 FF A9 CC 20 00 DD :1D
1780 06 CC 80 00 DD 08 CC 00 00 F7 FD 0F DC 06 93 00 :7B
1790 10 2C 00 1A 9E 06 4F E6 84 9E 08 E7 84 DC 08 C3 :6B
17A0 00 01 DD 08 DC 06 C3 00 01 DD 06 16 FF DE DC 08 :46
17B0 9E 02 ED 84 4F F6 FD 0F DD 0A DC 04 83 00 00 10 :BC
17C0 26 00 05 10 DE FE 35 FF 17 FD 8B DD 04 DC 04 36 :E1
17D0 06 CC 00 00 17 FD 6A 36 06 DC 00 36 06 CC 60 00 :D0
17E0 17 FD 3E A4 C0 E4 C0 10 27 00 37 DC 04 83 00 0D :38
17F0 10 27 00 15 DC 04 9E 00 E7 84 DC 00 C3 00 01 DD :B2
---------------------------------------------------------
Sum: 45 44 A7 CD B9 A2 79 D8 DA E3 0D 70 DA 2A 3F 15 :3B


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1800 00 17 FD 52 DD 04 16 FF E2 CC 00 00 9E 00 E7 84 :13
1810 DC 00 C3 00 01 DD 00 DC 0C DD 04 DC 0C DD 04 16 :25
1820 FF AB CC 20 00 DD 06 CC C0 00 DD 08 CC 00 00 F7 :AD
1830 FD 0F DC 06 93 00 10 2C 00 1A 9E 06 4F E6 84 9E :D2
1840 08 E7 84 DC 08 C3 00 01 DD 08 DC 06 C3 00 01 DD :83
1850 06 16 FF DE DC 08 9E 02 ED 84 4F F6 FD 0F DD 0A :26
1860 10 DE FE 35 FF 10 DE FE 35 FF 30 00 00 00 00 00 :70
---------------------------------------------------------
Sum: F6 AC E9 67 54 99 A8 D4 AD 4E DA E6 85 D2 4D 16 :D0
```

リスト6　vis2.Oマシン語ダンプ・リスト（$1400～$16FFは，リスト3と同じです）

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1700 34 7F 86 6F 1F 8B 5F 1F 03 10 DF FE CC 00 0D 17 :B0
1710 FF 4C CC 00 0A 17 FF 46 CC 6F FC DD 00 CC 00 00 :5D
1720 DD 02 CC 80 00 DD 04 CC 20 00 DD 06 9E 00 EC 84 :E9
1730 DD 08 CC 00 00 F7 FD 0F DC 04 36 06 DC 08 17 FD :C8
1740 E0 36 06 DC 06 36 06 CC 60 00 17 FD D4 A4 C0 E4 :96
1750 C0 10 27 00 1A 9E 04 4F E6 84 9E 06 E7 84 DC 06 :5D
1760 C3 00 01 DD 06 DC 04 C3 00 01 DD 04 16 FF C9 4F :59
1770 F6 FD 0F DD 0A CC 20 00 DD 0C DC 0C 93 06 10 2C :7B
1780 00 50 9E 0C 4F E6 84 36 06 CC 00 00 17 FD B2 36 :B7
1790 06 DC 0C 36 06 DC 06 17 FD 87 A4 C0 E4 C0 10 27 :E6
17A0 00 12 9E 0C 4F E6 84 17 FE B4 DC 0C C3 00 01 DD :C7
17B0 0C 16 FF CE DC 0C 93 06 10 2C 00 13 DC 0C C3 00 :6A
17C0 01 DD 0C CC 00 0D 17 FE 95 CC 00 0A 17 FE 8F 16 :FD
17D0 FF A8 DC 04 93 08 10 2C 00 74 CC 20 00 DD 06 CC :6D
17E0 00 00 F7 FD 0F DC 04 93 08 10 2C 00 1A 9E 04 4F :C5
17F0 E6 84 9E 06 E7 84 DC 06 C3 00 01 DD 06 DC 04 C3 :A5
---------------------------------------------------------
Sum: 3E 75 EB 74 62 1B 35 4B 5F 97 D5 E0 7B 1F A8 2B :27


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1800 00 01 DD 04 16 FF DE 4F F6 FD 0F DD 0A CC 20 00 :F9
1810 DD 0C DC 0C 93 06 10 2C 00 34 9E 0C 4F E6 84 83 :C0
1820 00 00 10 27 00 12 9E 0C 4F E6 84 17 FE 30 DC 0C :D9
1830 C3 00 01 DD 0C 16 FF E2 DC 0C C3 00 01 DD 0C CC :05
1840 00 0D 17 FE 19 CC 00 0A 17 FE 13 16 FF C4 CC 00 :DE
1850 00 17 FE 0A 10 DE FE 35 FF 84 00 00 00 00 00 00 :C3
---------------------------------------------------------
Sum: A0 31 DF 1C DE D7 89 A8 37 A5 07 16 57 83 58 5B :38
```

リスト7　viメイン・プログラム変更箇所

```
190 OPEN"I",#1,F$:LOADM"vil2.O",,R:CLOSE #1
210 OPEN"O",#2,"vi.out":LOADM"vis2.O",,R:CLOSE #2
```

RANDOM BOX

FM-7
BASICテキスト　文字別サーチ・プログラム

■COMPAC Sgn

このプログラムは，メモリ上にあるBASICテキストから，任意の文字列を含む行をみつけ，その文番号を出力するものです．ただし，中間言語化されたものは見出不可能です．

リロケータブルなので，好きな位置に移動して使ってください．他機種への変更は，表1のアドレスです．

使い方

“INPUT NAME?”と表示したら，見つけたい文字列（長さは255バイトまで可能）を入力して，最後に RETURN キーを押してください．その文字列を含む行番号が出力されます．終わるには，CTRL + C を入力してください．

大きなBASIC，Kコンパイラのプログラム変数名を，見つけるのに便利です．

なお，スタート・アドレスは先頭です．

表1

アドレス	内容	機能
$6003,4	$D08E	Aレジスタを画面に出力する．
$6006,7	$D807	画面から1行入力する．CTRL + C で処理を終わります．
$6008,9	$043D	256バイトの1行入力用のバッファ．
$600D,E	$0033	BASICテキスト先頭アドレスの入っているアドレス．

図1　出力例

```
INPUT NAME?BSR
00240 00260 00280 00490 00500 00520 00600 00620 01060
INPUT NAME?
```

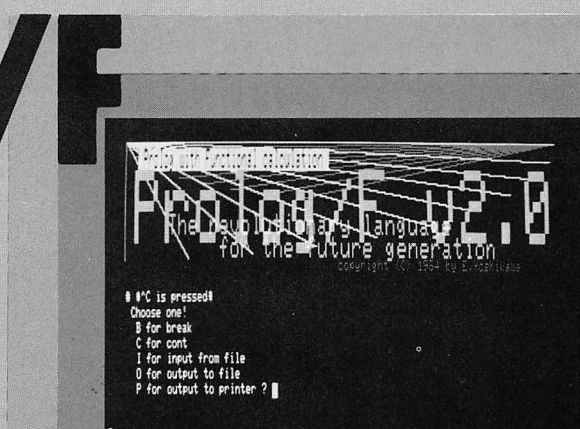
リスト1

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6000 20 08 7E D0 8E 7E D8 07 04 3D 34 7F BE 00 33 34 :7A
6010 10 8D 54 30 8D 00 44 17 00 8D AE 8C EB 8D E6 25 :53
6020 36 35 10 EC 84 27 E5 34 06 EE 02 30 04 10 AE 8C :9F
6030 D7 A6 A0 AC E4 27 1C A1 80 26 F8 A6 A0 27 06 A1 :43
6040 80 27 F8 20 E8 1F 30 30 8D 00 69 8D 23 8D 58 86 :37
6050 20 8D AF 35 10 20 CC 35 10 35 FF 49 4E 50 55 54 :96
6060 20 4E 41 4D 45 3F 00 86 0A 8D 97 86 0D 8D 93 39 :20
6070 34 56 32 7C ED 62 33 8C 24 C6 05 E7 E4 6F 61 EC :BC
6080 62 A3 C4 25 04 6C 61 20 F8 E3 C1 ED 62 A6 61 8B :5C
6090 30 A7 80 6A E4 26 E6 6F 84 32 64 35 D6 27 10 03 :7F
60A0 E8 00 64 00 0A 00 01 34 12 A6 80 26 02 35 92 17 :C9
60B0 FF 50 20 F5 00 00 00 00 00 00 00 00 00 00 00 00 :64
---------------------------------------------------------
Sum: AA 62 64 3A 9F 3E 94 2D E3 21 85 6C E9 9F 71 2A :60
```

FM-7

コンピュータとの新しい接し方のヒントを与えてくれる言語

Prolog/F V.2

東大人工知能研究会　吉川永一

技術が日々様変りする現代．過当競争の激しい社会．多忙な日々に，自らを見失いがちな中で，何か本当に信頼して相談できる相手，自分のブレイン的存在を夢見て，コンピュータのマニアになった私のような人が他にもきっといると思います．

しかし，実際にはコンピュータはごく機械的な作業に，用途が限られているのが現状です．解いて欲しい問題を解決するどころか，人間側にますます多くの問題を押しつけてきます．

もっと他のコンピュータの活用方法は，ないのでしょうか．本稿では，PROLOG/Fの使い方の説明を通じて，こういった問題について考え，僭越ながら，Prologの活用方法について例題を通し，論じていきたいと思います．

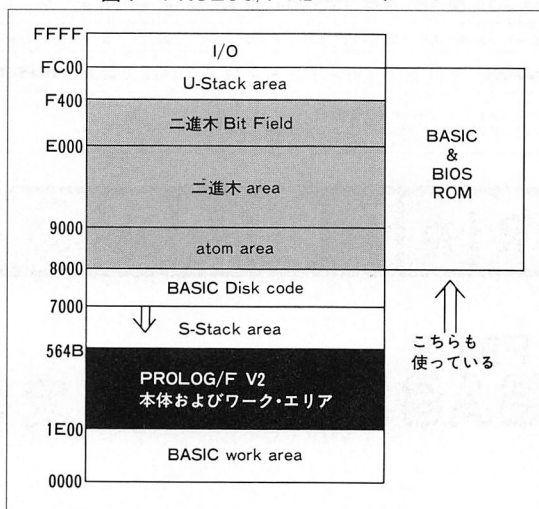
1 人工知能の探究の道具としてのProlog言語

PROLOG/F V.2は旧バージョン(PROLOG/F:I/O '83年8月号)に対し，ディスクの入出力を便利にしたり，メモリ容量を改善したものです．当バージョンの使い方や旧バージョンとの差異などの詳しい話は②にまわして，ここではPrologをどう使うかについて論じます．したがって，ここに出てくるプログラム例などを，実際に試しながら読みたい方は②の方を合わせてお読みください．なお，ここに出てくるプログラム例は，特に注意書きがない限り，新・旧バージョンで実行できます．

1.コンピュータの活用方法の一提案

従来のコンピュータの使い方は，問題となっている対象に対し，それとうまく対応づけられるような現象を，人間がコンピュータ内部に起こしてやり，それを観察するといったものでした．そのものずばりのシミュレーション関係のものから，事務用簡易ソフトまで，多くのものをこの観点で捕えることができると思います．このようなコンピュータの使い方は，大変意義のあることです．しかし，少しでも違う対象に対しては，適用できないプログラムしか作れないとすれば，場合場合に応じて人間がプログラムを作らなければなりません．

広い範囲の事物に対し，そして場合に応じて臨機応変に，有効に動作するようなプログラムには，何が必要なのでしょうか．その手懸りを次のように考えてみます．従来でも人間の大きな労力をコンピュータの動作の一部とみなせば，

図1　PROLOG/FV.2のメモリ・マップ

そのようなプログラムは既に実現していたとみなせます．すると，その人間の労力の部分さえ，コンピュータが独自で実行すれば，応用範囲の広い柔軟なシステムが得られる，と考えられます．

では，いままで人間が労力を注いでいた部分とは，いったい何なのでしょうか．対象に関する知識を扱い，推論を行なうレベル――そのレベルの作業はシミュレーションとは違って，対象の動きをそのまま反映するようなものではありません．

対象の様々な性質を象徴する記号群を，あちこちに移動させるような動作が，複雑にからみ合ってできている作業なのです．こういった対象レベルとは違う動作を，ここでは**メタ対象レベル**の動作と呼ぶことにします．

メタ対象レベルの動作の中で，人間は経験を，または言葉，文字を通して他人と伝達し合うことで，知識を蓄えます．この蓄えた知識で脳のハードウェアを補って余りあるほどの成果をあげてきたと思います．

とにかく，コンピュータに人間がやっているようなメタ対象レベルの知的作業をさせようと試みるのは，無駄なことではないでしょう．そして，その試みの1例として，数学的帰納法を使って，再帰的な思考プロセスを表現したProlog言語を作ってみました．以下に解説の意味も含めて，説明していきます．

2.コンピュータにいかに知識を表現するか

では，どうやったらコンピュータにメタ対象レベルの動作をさせることが，できるのでしょうか．いきなり人間の思考を，詳細に観察しようといっつも無理ですから，最初は思いきって，単純化して考えることです．

例として，次のような場合の人間の思考を，プログラム化することを考えます．いま，本が何冊か積んであるとします．本は一度に一冊しか取れないとします．このとき，一番下の本を取り出せと言われたなら，人間は上から一つずつ取れば良いというでしょう．

あたり前のようですが，このとき人間は次のような思考過程を辿っているはずです．本の山から一冊とることができる．とってみると残りは一冊減った山である．さて，変化したのは本の冊数だけであり，再び同じ動作ができるのは確かだろう．こうしていつかは本は一冊になり，それが目的の状態である……と．

実にこんな簡単な問題にも，人間は数学的帰納法を無意識のうちに用いているところを見ると，この思考法は非常に一般性があるのかもしれません．そこで，この数学的帰納法をコンピュータにやらせようというわけです．

さて，人間の思考をまねさせるにしても，いろいろなレベルが考えられるでしょう．思考の時間だけをまねして，〝人間がこれを考えると答を得るまで長くかかるから，走らせてから止まるまで長くかかるプログラムを作ればそれでいい〟といったのでは，しようがありません．

人間がこれを考えるには，〝ある情報を何度も想起しなければいけないから，プログラムも走らせると，**その情報に対応する**あるデータを何度もアクセスするようにすればよい〟．これは少し進歩したと言えます．しかし，注意しなければならないのは，人間が常に〝その情報に対応する〟といったような対応づけをしてやって，初めてコンピュータは知的な動作をした，と言えることです．こういった制限は，コンピュータが手足を持ち，自分の判断に基づいて実際に行動ができるようになるまでは，除去不可能な仮設でしょう．

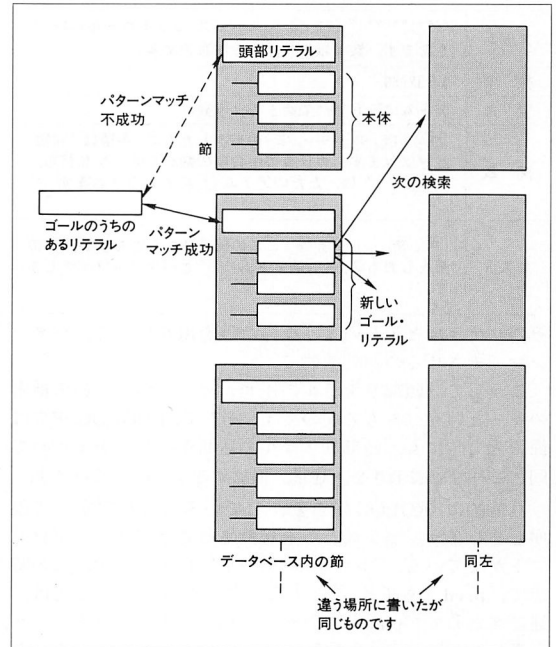
しかし，こういった仮設を前提の上で，人間の思考をまねすることもけっして意味のないことではありません．実際に行動する機能が付いたとき，プログラムにそのコマンドを追加するだけで，本当に知的な行動をするようになるなら，そのプログラムはこの拡張を加える前に，もう既に人間の思考を充分反映している，といえるのではないでしょうか．

このような考えに基づいて，人間の思考経路に対応させられるような動作をプログラムとして表現するには，どういうプログラムを書けば目的の動作が期待できるか，を知らなければなりません．そのプログラムは，インタープリタで実行されると，知的行動を生み出すもの，いわばコンピュータにとっての知識です．

そして，Prolog言語によって書かれた知識は，人間にとっての知識表現形態にも近く，わかり易いといった点で，Prolog言語はこうした用途に最適な言語であるといえます．

3.動作をProlog言語で表現する方法

Prologのプログラムとは，いくつかの**節**(clause)と呼ばれるデータを，データベースに蓄えることを意味します．それぞれの節は，1つ以上の**リテラル**でできています．そして，節の先頭のリテラルを**頭部**と呼んで，特に区別します．残りのリテラル（0個の場合もある）をまとめて，本体と呼びます．さて，Prologが動作状態にあるときは，常に

図2　Prologの動作

いくつかのリテラルが特別の状態にあります．それを**ゴール**と呼ぶことにします．

Prologの動作とは，次のように定式(**図2**)化できます．

『ゴールから一つリテラルを取り出し，そのリテラルとうまく適合する頭部リテラルを持つ節を，データベースから見つけ出す．それに成功したら，もとのリテラルをゴールから取り去る．そして，その節の本体のリテラルをすべてゴールに付け加える．ゴールが空になるまでそれを繰り返す．』

さて，これを見て変に思われる方もあるかも知れません．たとえば，次のようにです．上の動作の繰り返しのうち，一回の動作につき一個のリテラルがゴールから除去されますが，呼ばれた節の本体の複数のリテラルが追加されるのだから，いつまでたってもゴールは空にさらないのではないか？

実はここで，本体のない（本体のリテラルが0個の）節が重要な役割を演じます．この節が呼びだされたときだけ，ゴールのリテラルが減るわけで，こういう節を特に**事実**と呼びます．

さて，上に定義したPrologの動作の中に，〝適合〟という用語がでてきました．**適合（パターンマッチ）**とは，つき合わせてみて，同じ形をしている場合を言う，とまず考えてください．要するに，Prologの動作とは，〝あるリテラルと同じ形のリテラルを捜すこと〟よりなっているわけです．

では，そのリテラルの形とはどういったものでしょうか．それを説明するために，以下に**項**というものを定義します(**表1**)．

リテラルは，項のうち〝関数〟の形をしたもののみが許されます．リテラルの場合は，関数の引数が0個でも〝( )〟(カッコ）を省略しないでください．

さて，このように，リテラルはすべて関数ですから，関数名としてのアトムが必ずあるわけで，これをそのリテラルの**述語**と呼びます．リテラルとリテラルがパターンマッチするとき，もちろん述語同士も等しくなければなりません．従って，あるリテラルに対し，頭部リテラルの述語が，

表1　項の定義

アトム	"("")""["]"","".""-",スペース,コントロール・コードを含まず，数字以外で始まる任意の文字列.
数　字	0～32768.
変　数	アトムのうち"＊"で始まる文字列.
関　数	(アトム)(項,項,……,項)の形をしたもの. 各項は"別数"と呼ばれます. カッコの中の項の数が0個，すなわち，"(アトム)( )"は，ただのアトム(アトム)とみなされます.
リスト	[項,項,……,項項(ここの部分はなくともよい.)]の形をしたもの，[ ]はアトム"nil"とパターンマッチします.

そのリテラルと等しいような節だけを取り出して，パターンマッチさせるのが効率的です．

こうして『頭部リテラルの述語』というのは，節の検索のキーというべきものなのです．そして，PROLOG/Fでは節の入力時にも，頭部リテラルの述語が同じ節をまとめて同じモードで表示させ，修正，追加するようにしています．

具体的にPROLOG/Fを動かしていることを想定して説明しましょう．当システムを起動させると"#"のプロンプトがでている，"コマンド・モード"に入ります．この時点で，**pred**　述語名⏎と入力してください．たとえば，述語名として"plan"という文字列を入力してみましょう．行番号100が行頭に自動的に出力されて，再び入力をうながしています．

ここから，**plan**を頭部リテラルの述語として持つ節が入力されることは，**pred plan**⏎の入力でシステムに伝わっているので，節の頭部リテラルは述語を省略して，つまり"((項),…,(項))"の部分だけを入力するようにします．そしてそれにひき続き，本体を"—(リテラル)"の繰り返しで入力していきます．

さて節の入力のしかたはわかったが，いかにして動作を表現すればよいでしょうか．たとえば，"plan"という述語で，"目的のものを達成する方法を捜し出す行為"を表現するとします．特定の場合だけに働くのではなく，同じプログラムを色々な場合に働かせたいのであれば，何か自由度のある部分がなければいけません．そこで**変数**というものを使うことになります．こうした変数の効用は，どんなコンピュータ言語を，少しでも知った人ならわかっていると思います．

Prologの場合一風変わっているのは，変数は未代入のままでも良いということ，一度代入されると，その上にまた代入して前の値を書き換えられないということです．

変数が未代入のまま受けわたしされるというのは，奇妙に聞こえるかも知れません．"未確定"値が渡されるのではなく，伝えることがあったらここへ入れてくれ，とその変数の値を入れる**場所**が渡されていくと考えてください．

ここで，変数の受け渡しという話がでましたが，これは節から節へとパターンマッチの際に受け渡しがされるのです．前にパターンマッチとは，同じ形かどうか判断することと言いましたが，変数が一方のリテラルにあれば，対応する箇所のもう一方のリテラル内の項を代入して，同じ形になるようにするのです．

このように変数の受け渡しは，呼ぶ側，呼ばれる側どちらのリテラルにとっても対称な形をしており，他の言語の関数のように引数は呼ぶ側からわたすデータとは限らないのです．

そういうわけで，**plan**には3引数を持たせます．

図3　まずはごく単純化して——帰納法の骨組だけを表現

```
# pred plan
100 (*prc,*fnc,*proved)
110 -member([*prc,*fnc],*proved)
120 (*prc,*fnc,*proved)
130 -reducedCase(*fnc,*reducedFnc)
140 -resolve(*fnc,*fncs)
150 -plans(*prcs,*fncs,[[rprc,*reducedFnc].*proved])
160 -gather(*prcs,*prc)
170 ;
;
# pred plans
100 ([],[],*p)
110 ([*prc.*prcs],[*fnc.*fncs],*p)
120 -plan(*prc,*fnc,*p)
130 -plans(*prcs,*fncs,*p)
140 ;
;
# pred member
100 (*a,[*a.*x])
110 (*a,[*b.*x])
120 -member(*a,*x)
130 ;
;
# pred reducedCase
100 (*x,reduced(*x))
110 ;
;
# pred resolve
100 (A,[B,reduced(A),C])
110 ;
;
# pred gather
100 (*prcs,label(rprc,*prcs))
110 ;
;
# pred think
100 (*prc,*fnc)
110 -plan(*prc,*fnc,[
120  [prc1,B]
130  [prc2,C]
140 ])
150 ;
;
# exec
100 -think(*prc,A)
110 ;
;
!!!success!!!
think(label(rprc,[prc1,rprc,prc2]),A)
```


planの3変数

＊prc……見つかった手続き(値を返すための変数)
＊fnc……目的の機能
＊proved…どういう手続きをすれば，どういう機能が果たされるか，についてわかっていることすべて

という3変数を，頭部リテラルの引数として並べてみましょう．＊prcのように，こちらから結果として返す変数もちゃんと受け取っておかないと，結果を返すことができなくなります．

しかしいま，＊prcは返信用変数と決めてかかりました．しかし，できたプログラム（**図3**）は，いま定義している節を呼びだすリテラルの＊prcに対応する位置に変数でないものを入れ，＊fncの方を変数にして呼びだしても実行可能です．そして，＊fncの方に手続き＊prcを実行した結果が返されます．

このように，Prologではこのデータを入力すると，この出力が得られるように作ったつもりのプログラムが逆方向にもうまく働いて，ちょうど逆関数を適用したような結果が得られることがあります．

このことから，Prologのプログラムは，入力から出力へと変化させる関数を定義したものではなく，『入力と出力の"関係"を定義したもの』，といえるのです．

そして，この入出力を逆転しても，うまく双方が"関係"を満たす場合，カット・オペレータを使わなくても，無限ループにならずに終わります．

Prologのプログラムに慣れてくると，無意識のうちにこうした性質の恩恵を被っているものですが，カット・オペレータや無限ループが予期せぬバグを生むこともあり，やっかいです．しかし，知識の表現言語として，Prologのプ

図4　簡単な問題を帰納法を用いて解かせる

今回は目的の機能を，一つ一つの手続きに対応するものに分割して与えてしまわず，コンピュータが個々の手続を組み合わせ，目的の機能が果されるかどうか検索させている．

```
# pred gather
100 (*initPrc,*prcs,recursivePrc(*initPrc,*prcs))
110 ;
;
# pred think
100 (*prc,*fnc)
110 -plan(*prc,*fnc,[
120    [take1(*a),change([repeat(*a).*b],[dec(repeat(*a)).*b])],
130    [take2(*d),change([*c,repeat(*d)],[*c,dec(repeat(*d))])],
140    [put1(*e),change([repeat(*e).*f],[inc(repeat(*e)).*f])],
150    [put2(*h),change([*g,repeat(*h)],[*g,inc(repeat(*h))])]
160 ])
170 ;
;
# exec
100 -think(*prc,change([repeat(a),repeat(a)],[[],repeat(a)]))
110 ;
;
!!!success!!!
think(recursivePrc([],[take1(a),put2(a),rprc]),change([repeat(a),repeat(a)],[[],
repeat(a)]))
# pred plan
100 (*prc,*fnc,*proved)
110 -member([*prc,*fnc],*proved)
120 (*prc,*fnc,*proved)
130 -initCase(*fnc,*initFnc)
140 -plans(*initPrc,*initFnc,*proved)
150 -reducedCase(*fnc,*reducedFnc)
160 -plans(*prcs,*fnc,[[rprc,*reducedFnc].*proved])
170 -gather(*initPrc,*prcs,*prc)
180 ;
;
# pred plans
100 ([],change(*x,*x),*p)
110 ([*prc.*prcs],change(*obj,*obj3),*p)
120 -plan(*prc,change(*obj,*obj2),*p)
130 -plans(*prcs,change(*obj2,*obj3),*p)
140 ;
;
# pred member
100 (*a,[*a.*x])
110 (*a,[*b.*x])
120 -member(*a,*x)
130 ;
;
# pred initCase
100 (change([repeat(*x),repeat(*x)],[[],repeat(*x)]),
110  change([[],repeat(*x)],[[],repeat(*x)]))
120 ;
;
# pred reducedCase
100 (change([repeat(*a),repeat(*a)],[[],repeat(*a)]),
110  change([dec(repeat(*a)),inc(repeat(*a))],[[],repeat(*a)]))
120 ;
;
```

ログラム自体を研究手段に使っているのであって，インタープリタが実際に実行できるかどうかは重要でないという見方からすれば，純粋にPrologは関係を表わしているのだ，と言い切ってもかまわないでしょう．Prologプログラムは，コンピュータが読まなくとも，〝人工知能研究者〟が読めば，それだけでも意義があるのです．

さて，3引数と件に起動された節**plan**は，本体としてどういうリテラルを充足に向けてゴールに送りだせば，期待する**plan**の役割が果たされるのでしょうか．**リテラルの充足**とは，そのリテラルと適合する節が見つかり，その本体がない，あるいは本体のすべてのリテラルが充足されたときをいいます（再帰的定義．Prologを使う人はいち早く慣れてください!）．充足が終わったときに，3変数＊prc，＊fnc，＊provedはうまく**plan**の関係を満たすように代入されていなければなりません．

先に変数＊provedを決定したところで〝～すべて〟という表現をしましたが，こういう何かを寄せ集めたものは，どう表現すればよいでしょう．**リスト**はこういう物を表現する最適のデータ構造です．[〈知識1〉,〈知識2〉,……]のように，データをリストに格納することで，1つの項として1変数＊provedに受け渡すことができます．

リストを活用する方法の説明として，述語**member**を解説します．図3の**member**の定義を見てください．2引数目にリストがくるのですが，[＊a．＊x]となってい

図5　ある長さのもう少し難しい問題を帰納法を用いて解かせる．

問題　ある長さの棒　を，小さい順に並べた棒の列に加え，　小さい順壊さぬようにするには．

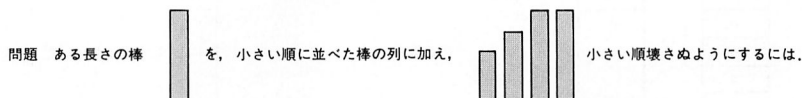



図4の方法をそのまま用いるために，かなり単純化してヒントを与えている．

```
# pred plan
100 (*prc,*fnc,*proved)
110 -member([*prc,*fnc],*proved)
120 (*prc,*fnc,*proved)
130 -initCase(*fnc,*initFnc)
140 -plans(*initPrc,*initFnc,*proved)
150 -reducedCase(*fnc,*reducedFnc)
160 -plans(*prcs,*fnc,[[rprc,*reducedFnc].*proved])
170 -gather(*initPrc,*prcs,*prc)
180 ;
;
# pred plans
100 ([],change(*x,*x),*p)
110 ([*prc.*prcs],change(*obj,*obj3),*p)
120 -plan(*prc,change(*obj,*obj2),*p)
130 -plans(*prcs,change(*obj2,*obj3),*p)
140 ;
;
# pred member
100 (*a,[*a.*x])
110 (*a,[*b.*x])
120 -member(*a,*x)
130 ;
;
# pred initCase
100 (change([elem,order],extOrder),
110  change([next,order],extOrder))
120 ;
;
# pred reducedCase
100 (change([elem,order],extOrder),
110  change([elem,dec(order)],dec(extOrder))
120 ;
;
# pred gather
100 (*initPrc,*prcs,recursivePrc(*initPrc,*prcs))
110 ;
;
# pred think
100 (*prc,*fnc)
110 -plan(*prc,*fnc,[
120    [take,change([elem,order],[elem,dec(order)])],
130    [put1,change([next,order],extOrder)],
140    [put2,change(dec(extOrder),extOrder)]
150 ])
160 ;
;
# exec
100 -think(*prc,change([elem,order],extOrder))
110 ;
;
!!!success!!!
think(recursivePrc([put1],[take,rprc,put2]),change([elem,order],extOrder))
```

棒の列の頭から1つずつ棒を取り除いていけば，いつかは加えたい棒が棒の列の先頭より小さくなります．これを，棒がelemからnextに変わったと表現しているのです．

ると，＊aにはそのリストの先頭の要素，＊xには2番目以降のリストが入るのです．

1番目の節は本体(－リテラルの形のもの)が一つもない節ですから，これにパターンマッチすれば **member**の関係『等1引数は第2引数であるリストの要素である』を満たしているはずです．

(＊a, [＊a. ＊x]) は第1引数が第2引数の先頭の要素であるから，確かに成立しています．2番目の節は第2引数を[＊b. ＊x]という形で受け取って，同じ **member** に第2引数として＊xを渡しています．これは，先頭の要素をやりすごしたに他なりません．よって **member** の動作は，次のように表現できます．『第1引数をリストである第2引数の先頭の要素とのパターンマッチを試みよ．失敗したら第2引数の先頭要素を取り去って，再び **member** を呼び出せ．』これが **member** に期待した動作であることは確かです．

話は **plan** に戻ります．変数＊provedに代入するリストの要素として，2要素のリスト［手続き，その機能］を入れることにします．こうすれば，**plan** の定義は図**3**のように，**member** を呼び出せばそれで終わりです．

plan の2番目は節の何かというと，これが帰納法を意識したものなのです．帰納法の仮定，『より簡単な場合には，既に求める手続きは与えられている』ことを＊provedに付け加えて **plans** を呼び出しています．これ以上は図**3**をじっくり眺めれば，わかるでしょう．

plans は複数の手続きの集まりを見つけるところだとだけ言っておきます．

Prologの実行を開始させるには，コマンド・モードから **exec**↵し，次に"－(リテラル)"の集まりを書いてください．これは，ゴール節と呼ばれ，その中のすべてのリテラルを，ゴールとして実行が開始されます．

図6　図5を完全にする

図5の問題を，もっと完全な証明を作らせつつ，解かせている．このプログラムを旧バージョンで走らせるためには，図7の変更が必要です．

```
# pred induction
100 (obj(*s,*sProp),obj(*p,*pProp),*proved)
110 -initCase(*s,*initHypo)
120 -oncePlans(*initPrcs,obj(*s,[*initHypo.*sProp]),obj(*p,*pProp),*proved)
130 -ｷﾉｳﾎｳﾉ・ｼｭｯﾊﾟﾂ・ﾂｳｶ()
140 -induction()
150 (obj(*s,*sProp),obj(*p,*pProp),*proved)
160 -reducedCase(*s,*reducedPrc,*reducedHypo)
170 -oncePlans(*prcs,obj(*s,[*reducedHypo.*sProp]),obj(*p,*pProp),
180    *reducedPrc)
190 ;
;
# pred plan
100 (*prc,obj(*s,*sProp),obj(*p,*pProp),*proved)
110 -knowledge(*prc,obj(*s',*s'Prop),obj(*p,*pProp),*proved)
120 -included(*s',*s)
130 -inferred(*s'Prop,*sProp)
140 ;
;
```

（*s：棒や棒の列などの，現在までの操作で得たもの．　*sProp：それらの間で成立する関係．）

```
# pred plans
100 ([],obj(*s,*sProp),obj(*p,*pProp),*proved,*x)
110 -included(*p,*s)
120 -inferred(*pProp,*sProp)
130 ([*prc.*prcs],obj(*s,*sProp),obj(*p,*pProp),*proved,rprc)
140 -plan(*prc,obj(*s,*sProp),obj(*p',*p'Prop),[])
150 -append(*p',*s,*s')
160 -append(*p'Prop,*sProp,*s'Prop)
170 -ﾂｳｶ(*prc)
180 -plans(*prcs,obj(*s',*s'Prop),obj(*p,*pProp),*proved,[])
190 ([*prc.*prcs],obj(*s,*sProp),obj(*p,*pProp),*proved,*x)
200 -plan(*prc,obj(*s,*sProp),obj(*p',*p'Prop),*proved)
210 -append(*p',*s,*s')
220 -append(*p'Prop,*sProp,*s'Prop)
230 -ﾂｳｶ(*prc)
240 -plans(*prcs,obj(*s',*s'Prop),obj(*p,*pProp),*proved,*prc)
250 ;
;
```

（130～180：再帰的な呼び出しを2度以上使わないようにするための，ちょっとした手助けです．これがないと無限ループになってしまう．）

```
# pred oncePlans
100 (*prc,*obj1,*obj2,*proved)
110 -plans(*prc,*obj1,*obj2,*proved,[])
120 -!()
130 ;
;
# pred member
100 (*a,[*a.*x])
110 (*a,[*b.*x])
120 -member(*a,*x)
130 ;
;
# pred included
100 ([],*y)
110 ([*a.*x],*y)
120 -member(*a,*y)
130 -included(*x,*y)
140 ;
;
# pred inferred
100 ([],*y)
110 ([*a.*x],*y)
120 -member(*a,*y)
130 -inferred(*x,*y)
140 ([*a.*x],*y)
150 -infKnowledge(*a,*z)
160 -included(*z,*y)
170 -inferred(*x,*y)
180 ;
;
```

（e(*e)，l(*l)：変数の“型付け”…変数に代入されるものが棒であるのか，棒の列であるのかを示す．これがあると無駄な探索が省ける．）

```
# pred infKnowledge
100 (next(e(*e),l(*l)),
110   [next(e(*e),l(*l')),cns(e(*e),l(*l'),l(*m)),notNext(e(*f),l(*m)),
120    cns(e(*f),l(*l'),l(*l))]
130  )
140 (cns(e(*e),l(*l),l(*l')),
150   [cns(e(*f),l(*m),l(*l)),cns(e(*f),l(*m'),l(*l')),cns(e(*e),l(*m),l(*m')]
160  )
170 ;
;
# pred append
100 ([],*x,*x)
110 ([*a.*x],*y,[*a.*z])
120 -append(*x,*y,*z)
130 ;
;
```

図6 続き

```
# pred initCase
100 ([e(*e),l(*l)],next(e(*e),l(*l)))
110 ;
;
# pred reducedCase
100 ([e(*e),l(*l)],
110  [rprc,
120    obj([e(*e),l(*m)],[cns(e(*x),l(*m),l(*l))]),
130    obj([l(rprc(*e,*m))],[cns(e(*e),l(*m),l(rprc(*e,*m)))])
140    ],
150  notNext(e(*e),l(*l))
160 )
170 ;
;
# pred knowledge
100 (*prc,*obj1,*obj2,[*prc,*obj1,*obj2])
110 (put,
120    obj([e(*e),l(*l)],[next(e(*e),l(*l))]),
130    obj([l(put(*e,*l))],[cns(e(*e),l(*l),l(put(*e,*l)))]),
140    *proved)
150 (take,
160    obj([l(*l)],[]),
170    obj([e(take1(*l)),l(take2(*l))],
180     [next(e(take1(*l)),l(take2(*l))),cns(e(take1(*l)),l(take2(*l)),l(*l))]),
190    *proved)
200 ;
;
# exec
100 -induction(obj([e(e),l(l)],[]),
110    obj([l(*l)],[cns(e(e),l(l),l(*l))]),[])
120 ;
;
ﾂｳｶ(put)
ｷﾉｳﾎｳﾉ・ｼｭｯﾊﾟﾂ・ﾂｳｶ
ﾂｳｶ(take)
ﾂｳｶ(rprc)
ﾂｳｶ(put)
!!!success!!!
induction(obj([e(e),l(l)],[]),obj([l(put(take1(l),rprc(e,take2(l))))],[cns(e(e)
l(l),l(put(take1(l),rprc(e,take2(l)))))]),[])
```

construct…*lは*mに*xをつけ加えたものであるということ.

putが適用できるための条件…*eが列*lの先頭より小さければ列*eを加えても順序はくずれない.

(cons(car l)(rprc(再帰呼び出し e(cdr l)))と見ればそのままLispプログラム!

図7 旧バージョンで図6のプログラムを走らせるための変更箇所

アドレス	旧	新
21C0	C0 →	D0
25CA	C0 →	D0
25E2	9C →	AC
2623	C0 →	D0
2632	9C →	AC
266C	C0 →	D0
2672	CC →	E0
2CE2	9C →	AC

この変更によって，旧バージョンの二進木領域が12Kバイトから16Kバイトになります．因みに，V.2は20Kバイトです．

2 当システムの使用法

起動法

当システムはマシン語の本体，およびBASICで書かれた付加的な部分から成っています．起動は，まずBASIC部分をRUNさせるところから始まります．このとき，カセットあるいはディスケットの中にマシン語の本体が，ファイル名"PROLOGF2"でセーブされていなければなりません．

さて，RUNさせると"PROLOG/F"のタイトルがでた後，プロンプト"#"が出て，コマンド・モードになります．以下，コマンド・モードで使用できる命令について解説します．

図8 簡単なプログラム例(任意桁の2進数の足し算)

```
# pred add_1_bit
100 (0,0,0,0)
110 (0,1,1,0)
120 (1,0,1,0)
130 (1,1,0,1)
140 ;
;
# pred add
100 ([],[],[],0)
110 ([*a.*x],[*b.*y],[*c.*z],*carry)
120 -add(*x,*y,*z,*carry_below)
130 -add_1_bit(*a,*b,*d,*carry')
140 -add_1_bit(*d,*carry_below,*c,*carry'')
150 -add_1_bit(*carry',*carry'',*carry,*dummy)
160 ;
;
# execn
100 -add([1,0,1],[0,0,1],*x,*y)
110 ;
;
!!!success!!!
add([1,0,1],[0,0,1],[1,1,0],0)
# execn
100 -add([1,1,1,1,1],[0,0,0,1,0],*x,*y)
110 ;
; 同じプログラムで引き算も行なえるところがPrologの味のあるところです.
!!!success!!!
add([1,1,1,1,1],[0,0,0,1,0],[0,0,0,0,1],1)
# execn
100 -add([1,0,0,1,1],*x,[1,1,1,1,1],*y)
110 ;
;
!!!success!!!
add([1,0,0,1,1],[0,1,1,0,0],[1,1,1,1,1],0)
```

まず1桁の足し算を定義

add-1-bitを使って任意桁の足し算を，帰納的に定義する．

コマンド・モードで使える命令

❶predおよびfuncコマンド

pred［述語名］，あるいは**func**［関数名］でその述語，関数の定義をエディットすることができます．いずれの入力

でもエディット・モードに入ります（カラーディスプレイの場合には文字が青くなる）.

このモードに入ると，新たに定義する述語（関数）であれば行番号100をプロンプトして表示して入力モードに，また，すでに定義された述語（関数）であれば，その定義をすべて出力してから，次の行番号をプロンプトとして表示して入力モードとなります.

この行番号はエディットのために便宜的に使われているもので，一つの語を途中で切るようなことがなければ，いつでも次の行に移ってかまいません．↵キーで次の行に移れます．このエディタの使い方は，BASICのエディタとほとんど同じで，変更，挿入，削除が簡単に行なえます.

このエディット・モードに入ったときは，初めはBASICのAUTOモードに当たるように行番号を出力してきます．;↵でこのモードから抜け出せます．一連のエディットが終了したら，;↵で該当する述語(関数)のエディットから抜けられます.

さて，述語および関数の定義の書き方ですが，注意して欲しいのは，当システムではS表現ではなく，メタ表現を採用しているということです．つまり，述語にしても関数にしても，その引数は丸カッコで囲って，述語名あるいは関数名の後に書くということです．それから，一つの述語または関数をまとめてエディットする方針をとっているので，その述語(関数)の定義として，複数の節を与えるとき，各各の節の頭部(帰結句)に同じ述語名(関数名)を繰り返し書く必要はありません．つまり，各節の頭部は，引数のリストだけを書くようになっているので注意してください.

条件句はその引数リストに続けて，〝−〟(マイナス)記号の後に書けば良いのです.

以上，プログラム名を見ながら，自分でいろいろやってみてください.

❷execnコマンド

execn↵の入力によって，**goal**節の入力をするモードになります．**goal**節の句はすべて，〝−〟(マイナス)記号の後に続けて入力してください.

↵の入力により入ったモードも，**pred**，**func**コマンドのときと同じエディット・モードで，2回の;↵の入力により実行が始まります．**goal**節の入力にもエディット・モードとは少し奇異ですが，実は次に述べる**exec**コマンドの活用により，非常に便利なものとなっています.

❸execコマンド

このコマンドは**execn**コマンドと殆んど同じですが，唯一違うのは，前回入力したgoal節が残っているということです．これにより，前回実行したものの一部をエディタの機能で変更し，また実行するということができます.

❹traceコマンド

trace↵とすると，トレース・モードとなります．ただし，すでにトレース・モードのときは，そのモードが解除されます.

トレース・モードのときに，**exec**あるいは**execn**コマンドで**goal**節の実行が行なわれると，新たに節が選択されるたびにそれが出力されます.

❺listコマンド

list↵で，現在定義されている述語名，および関数名がすべて出力されます．これはV.2で新しく追加した機能です.

エスケープ・モード

V.2で新しく追加されたモードとして，〝エスケープ・モード〟があります．このモードは，コマンド・モードから

図9　Lazy-Evationを活用した素数計算

```
# func from
100 (*x)
110 -[*x.from(*y)]
120 -ex([*x,1,+,=:,*y])          —*xに1を足したものを*yに代入する
130 ;
;
# func filter
100 (*x,[*e.*1])
110 -filter(*x,*1)
120 -ex([*e,*x,mod,0,=])         —*eを*xで割った余りは0に等しい.
130 (*x,[*e.*1])                  この条件が満たされているとき，fil-
140 -[*e.filter(*x,*1)]           ter(*x,[*e.*l])はfilter(*x,*l)
150 ;                             に置き換えられる.
;
# func sieve
100 ([*e.*1])
110 -[*e.sieve(filter(*e,*1))]
120 ;
;
# pred print_all
100 ([*a.*x])
110 -ｿｽｳ(*a)                     未定義の述語が現われると，そのま
120 -print_all(*x)                ま印字するので，出力に利用できる.
130 ;
;
# exec
100 -print_all(sieve(from(2)))
110 ;
;
ｿｽｳ(2)
ｿｽｳ(3)
ｿｽｳ(5)
ｿｽｳ(7)
ｿｽｳ(11)
ｿｽｳ(13)
ｿｽｳ(17)
ｿｽｳ(19)
ｿｽｳ(23)
```

CTRL+C あるいは，CTRL+X を押すことによります．ディスクとの入出力を，旧バージョンよりも大幅に便利にしています.

使用法は画面に出るメニューに従えばよいわけです．ここでは注意点のみ書いておきます.

i↵**<ファイル名>**

i↵**<ファイル名>**，**o**↵**<ファイル名>**↵，**p**↵とも，実行後エスケープ・モードのままで，**c**↵を実行しなければなりません．これはたとえば，ディスクから入力しながらそれをプリンタに出力するなど2つ以上の操作を同時に指定したい場合を考えてのことなのです．また，**o**↵**<ファイル名>**↵**p**↵を指定すると，その後は画面に出力されるものが，そのままディスクまたはプリンタに出力されます．これを止めるには，再びエスケープ・モードに入って，**o**↵，**p**↵を実行すればよいのです.

組み込み述語

当システムの組み込み述語は〝ex〟と〝！〟(カットオペレータ)だけですが，ex述語の極めて多彩な機能のために，たいへん強力になっています．なお，当システムでは，実行中に未定義の述語に出合うと，その述語名ごと出力するという機能があります.

ex述語の使用法

ex述語とは，数値演算やBASICとのリンクその他，様々な機能を利用したいときに使う述語で，このPROLOG/Fの開発に使われた言語（FIDIAと名付けた，これも私の開発したもので，いずれどこかで発表しようと思っています)の機能を呼び出すものとなっています.

使い方は図9の**from**の定義に見られるように，逆ポーランド方式，すなわちデータを先に書き，演算を後に書く方式になっています.

代入は〝=:〟の次に変数を書けば良いのです．ここには別に変数以外のものがきても良く，その場合，左辺とのパターンマッチングに失敗すると，**ex**述語の呼び出し自体が失敗して，バックトラックを起こします.

ex述語の呼び出しが失敗し，バックトラックするのは，こ

図10 ex述語のバックトラック機能活用例(述語"+"の定義)

```
# pred +
100 (*x,*y,*z)
110 -ex([*x,*y,+,=:,*z])
120 (*x,*y,*z)
130 -ex([*z,*x,-,=:,*y])
140 (*x,*y,*z)
150 -ex([*z,*y,-,=:,*x])
160 ;
;
# execn
100 -+(5,1,*x)
110 ;
;
!!!success!!!
+(5,1,6)
# execn
100 -+(80,*x,99)
110 ;
;
!!!success!!!
+(80,19,99)
# execn
100 -+(*x,3,10)
110 ;
;
!!!success!!!
+(7,3,10)
```

図11 カット・オペレータの使い方

```
# pred NOTEQ                 NOTEQの誤まった定義
100 (*x,*x)          ――――2つの要素が等しいことを示す定義
110 -NOTEQ()                 の始まり
120 (*x,*y)          ――――上の条件が成り立つときの条件
130 ;                ――――2つの要素が等しくないことを示す
;                            別の定義
# execn
100 -NOTEQ(A,A)
110 ;
;
NOTEQ(A,A)***If:
-NOTEQ.              ←――――バックトラックはここで起こる

NOTEQ(A,A)**IsTrue

!!!success!!!
NOTEQ(A,A)
# pred NOTEQ
100 (*x,*x)                  NOTEQの正しい定義
110 -!()
120 -NOTEQ()
130 (*x,*y)
140 ;
;
# execn
100 -NOTEQ(A,A)
110 ;
;
NOTEQ(A.A)***If:
-!,and
-NOTEQ.

!!!failure!!!
NOTEQ(A,A)
```

のときだけではありません．"=:"が左にないとき，未確定の変数がある場合にも，バックトラックします．つまり，まだ確定していない値を利用しようとしたときです．図10を見てください．ここで定義された述語"+"は，3つの引数のうちどれが未確定でも，うまく"+"の関係を満たすべて変数に値を代入してくれます．

さて，バックトラックを故意に引き起こすことも可能で，す．もう一度図9を見てください．関数**filter**の定義中で使われている**ex**述語には，最後に"－"がありますね．この"－"というwordは，二つのデータを消費し，それが等しいときに真(16進数でFFFF)を返し，違えば偽(0)を返すwordです．

この他">"，"<"などの大小関係を示すwordがあります．また，**mod**というwordは前の二つのデータの前を後で割った余りを返します．この他，"+"，"－"，"*"，"/"などの四則演算wordが使えます．

さて，**ex**述語の最後にある"="のワードが，偽を返したとしましょう．すると，バックトラックが起きて，**filter**関数のもう一つの別の定義へと移ります．このように，**ex**述語内で行なった条件判断をPrologの動作に影響させることが可能です．つまり，**ex**述語は単に外部の手続きを呼びだすだけではなく，それらを組み合わせて，他の述語と同じように，場合により真，偽を返す述語を自由に構成することが可能です．

ex述語にはその他にも機能があります．"]"からはみ出した部分は，バックトラックの際に実行される物です．BASICなどとリンクさせるときにProlog本来のバックトラック機能では，変数の束縛は戻してくれるが，外部に対しておよぼした影響は元に戻さないため，バックトラックが不完全になってしまいます．そこでこの機能を使えば，外部におよぼした影響を元に戻すような手続きを，こちらから与えることができ，うまく解決できるわけです．

このシステムでは，極めて容易にBASICプログラムと結合できます．画面処理や実数演算，グローバル変数，配列の利用などにBASIC部分を活用すれば良いでしょう．

BASICとPROLOG/F V.2の値のやり取りは，BASIC側からは**USR1**で，PROLOG/Fからは"**basic**"というwordで行ないます．このwordは1つのデータを消費し，それをBASICに渡し，BASICから受け取った値を返します．

また，リアルタイムの入力用wordとして"**getword**"があります．

カット・オペレータ

ある条件句にどの節をマッチングさせるか――ここに選択の余地があり，その選択によって結果が変わらぬようにするのが，バックトラックであったわけです．しかし，逆にこの選択というのを有効に利用したいということもあるのです．カット・オペレータとは，このような目的のためにバックトラックの機能を局所的に制限する働きを持つ述語です．

図11を見てください．これは，ある述語が定義されているときに，その否定を定義するための一般的な手法を示した物です．"**NOTEQの正しい定義**"と"**誤った定義**"の実行過程を比較してください．誤った方では一番目の**NOTEQ**の節の三番目の句"**NOTEQ**"で不都合がおこると，バックトラックが起こり，二番目の**NOTEQ**の節を試みますが，正しい方では，別の**NOTEQ**の節は試みずに失敗してしまいます．

このように，**カットオペレータとはその句より後で不都合が起こって，バックトラックしようとしたときに，それをさせずに，その句を含む節を呼び出した条件節自身を失敗させてしまうという働きを持っています．**

つまり，カットオペレータのある節を選択すると，他の節を試みる機会が奪われる可能性もあるわけで，Prologが節を選択する順序は，入力された順にしたがうわけですから，どういう順で節を入力するかがこの場合，重要になります．

図12　パターン・マッチングの拡張

	プログラム上	内部表現	パターン・マッチの結果 PROLOG/Fにおいて	パターン・マッチの結果 従来のPrologでは
変数同士	*x	*x	*x＝*yを保つように互いに束縛される.	同　左.
	*y	*y		
変数とリスト	*x	*x	*xに［a, *y］が代入される.	同　左.
	[a, *y]	a *y		
変数と関数	*x	*x	*xにf(a, *y) が代入される.	同　左.
	f(a, *y)	f a *y		
リストとリスト	[a, *x]	a *x	個々の要素のパターン・マッチングに還元される.	同　左.
	[*y, b]	*y b		
関数と関数	f(a)	f a	関数名同士と引数同士のパターン・マッチングに還元される.	同　左.
	f(*x)	f *x		
関数とリスト	f(a)	f a	関数 f の定義があればその頭部とf(a) とをパターン・マッチングし, それに成功すれば, その本体と［a, *x］とをパターン・マッチングする.	パターン・マッチングに失敗する.
	[a, *x]	黒と白の衝突 a *x		

☆上図で定数a, bの内部表現が a b のようになっているのは, 定数が0引数の関数であるとみなされるからです.

☆すでに何かに束縛された変数は, その束縛値の方を上図に示しました. したがって上図において変数とは, 未束縛の変数のことです.

3 論理型言語と関数型言語の融合

以下, PROLOG/Fの関数機能について説明します.

きっかけ

Prologでは, データのやりとりは必ず変数を介して行なわなければならず, 関数型言語の持つ簡素性が失なわれているという難点が挙げられます[1]. もちろんそのために, 関数型言語とは違って, 返す値を複数個とれるなど, プログラムの自由度が上がったことは無視できません. しかし, 関数として捉え方が自然であるようなものを述語に表現たするには, 引数をもう一つ増やし, それを値（もともと関数の値として意識されたもの）を返すために使うなどと回りくどいことをしなければならず, プログラムの読みやすさの上からも改善すべき点のような気がしました.

関数評価も部分的に取り入れるにはどうしたらよいかと考えているとき, Prolog処理系で広く使われている表記法からあるヒントを得ました.

多くのシステムでは, 項が複数の語からなっているときは, 最初の語をカッコの外側に出して, たとえば$f(*x, *y, a)$ のように関数風に表記されます. またそういうシステムでは, 別の表記法も許し, $[f, *x, *y, a]$ のようにかぎカッコの場合はS表現風に表わします. しかし, このように2通りの表記法を許し, また, お互いにパターンマッチはしない（つまり両者を別のものとする）とはしているものの, それ以上の機能の違いはないようです. すなわち, いずれにしろパターンとしての意味しかなく, 関数風に表わされていてもそれを評価するわけではない, ということです.

せっかくこのように2通りの表記法を許したのなら, 前者(丸カッコを使って関数風に記述したもの)に対しては,

図13　関数の評価

```
# func alphabet
100 ()
110 -[A,B,C]
120 ;
;
# pred always_true
100 (*x)
110 ;
;
# pred A_is_member
100 ([A.*x])
110 ([*a.*x])
120 -A_is_member(*x)
130 ;
;
# execn
100 -always_true(alphabet)
110 ;
;
!!!success!!!
always_true(alphabet)
# execn
100 -A_is_member(alphabet)
110 ;
;
!!!success!!!
A_is_member(alphabet)
# func alphabet
100 ()
110 -[B,C,D]
120 ;
;
# trace
# exec
100 -A_is_member(alphabet)
110 ;
;
A_is_member(alphabet)***If:
/A_is_member([C,D]).
A_is_member([C,D])***If:
/A_is_member([D]).
A_is_member([D])***If:
/A_is_member(nil).
!!!failure!!!
A_is_member(alphabet)
```


述語の一連の定義とは別に関数定義を与えてやり, 実際に関数計算をすればよいのではないか, と考えたのがPROLOG/Fの出発点でした.

評価をパターン・マッチング時に

このようにして関数をPrologのプログラム内に表現する方法はわかったのですが, ではいったいコンピュータは, いつその関数の値を計算(評価)するべきなのでしょうか. 新たな節が起動される（つまりパターンマッチングの相手として選ばれる）ごとに, その節内の関数はすべて評価ずみの値と置き換えるという方法では, あまりに工夫がないように思えます.

従来のPrologにおいて, 実質的に計算が行なわれるところは一連の処理過程のどこであると言えるでしょうか. それは, パターンマッチングで収ると思います. なぜなら, リストをばらしたり, 再構成したりを繰り返しつつ, 結果を構成していく動作はPrologにおいてはすべてパターンマッチング時に行なわれるからです. したがって, 関数計算もパターンマッチング時に行なうのが自然であろう, という考えを押し進めていくうちに, 図12のような考えに到達しました.

節の内部表現の二進木には, 各ノードにすべて1ビットのフラグ（図では黒か白）を持たせました（HIPER LISP[2]にヒントを得た). このフラグの意味は, もし黒ならばそこから先は関数であり, 評価を必要とするという意味を持ちます. 白ならば, そこから先は確定しており, 置き換えられることはないという意味です. さて, パターンマッチングで, 白と黒がぶつかったとき, 白いノードはそこから先がリストですから, 個々の要素のパターンマッチングに還元されるべきところですが, もう一方が未確定とあっては先へ進めません. そこで, その時点で関数評価が始まるの

です．

関数評価は関数部分を，それと等しい(とされた)確定値と置き換えることを意味します．このときの引数受け渡し機構においても，パターンマッチングを使うことにしました．すなわち関数定義は次のように書きます．

double _ first([*a, *x]) = [*a, *a. *x]

これで**double_first**という関数が定義されたわけです（当システムにおいては〝=″記号のかわりに〝_″ SHIFI + ろ を使いますが，意味的には〝=″だと考えてください)．つまり，**double_first**で始まる項がプログラム中にあったとき，PROLOG/F実行時に，必要に応じて右辺のように変換されるのです．

パターンマッチングの自然な拡張になっていることに注目してください．従来のPrologでは，パターンマッチング時に行なえる基本的な演算は，分解と構成（リストと［*a, *x］のマッチングで，*a，*xに入る値，あるいは*a, *xにすでに値が束縛されているときに[*a, *x]を作る動作を考えてください)だけだったわけですが，PROLOG/Fでは関数をあらかじめ定義しておけば，分解，構成に加え，バラエティに富んだ動作が可能になります．

たとえば〝しっぽとり出し″をさせるには，

last _ fetch([*a]) = [*a] **last _ fetch([*a, *x]) = last _ fetch(*x)**

と定義して**last_fetch(*x)**と使えばよいのです．

もう一つ重要なことは，lazy-evalvationの実現にもなっていることです．いま，**alphabet**という関数（0引数）を次のように定義したとします．

alphabet ( ) = [A, B, C]

さて，次の2つの述語の定義があるとします．

①**always_true (*x)**

②**A_is_member ([A, *x])**

A_is_member ([*a, *x]) = A_is_member (*x)

①は*xがなんであっても常に真．②はリスト中に**A**という要素があれば真になります．ここで，

①′ー**always_true (alphabet)**

②′ー**A_is_member(alphabet)**

を実行させると，いずれも真になりますが，関数評価が行なわれるのは②′の場合だけです．これは①の述語は*xの内部がどうなっていようと，影響されないのに対して，②の方はリストの要素を問題にする述語だからです．それは〝値が必要となったとき，初めて計算をする″，言い換えれば，lazy-evaluationを行なっているからなのです．実際に実行したようすを図13に掲げておきます．

lazy-evaluationによって，無駄な計算が省かれるという利点がありますが，もっと顕著に効果が表れるのは，無限長のリストが扱えるようになるということです．たとえば，図9を見てください．**from (*x)** の値は*x，*x+1，*x+2,……という無限個の要素を持つリストです．もし，**from (*x)** の値を完全に求めてから……としていたら，素数が1つも求まらぬうちにオーバーフローするでしょう．

この素数を求めるアルゴリズムは，『エラトステネスのふるい法』を，ごく自然に記述した物となっています (bit '82年12月号[3]参照)．このようにPROLOG/Fは，計算機のぎこちなさを相当おおい隠し，人間に近い表現ができると言えるでしょう．

関数評価とバックトラック

もう一度図9を見てください．関数定義の中には〝ー″記号が2つある物があります．1番目の〝ー″の右に続く物が関数の値であることは先程話しましたが，では2番目の〝ー″の後は何でしょうか．

ここには述語を先頭に持つ句（条件句）を書き，意味的には『その命題が真であれば，この関数定義が有効である』ということです．必要ならば〝ー″記号を繰り返し使って条件句を複数個書いてもかまいません．それらの論理積が条件となります．

条件が成立しない場合には，バックトラックして他の関数定義を試します．たとえば図9の関数**filter**の定義を見てください．100行目と130行目とから始まる2つの定義があり，前者は*eが*xの倍数であるとき，後者はそうでないときに適用されます．ちなみに**filter**の意味を述べておくと，2つの引数を持ち，第1引数は数，第2引数は数のリストで，返す値は第2引数のリストから，第1引数の数の倍数をすべて除いたリストです．

このように関数評価の際にもバックトラックが行なわれ，また条件句の部分にカットオペレータ〝!″述語があればそれも有効です．ただし，バックトラックは，その関数評価が引き起こされたパターンマッチングが成功するまで有効で，その後はバックトラックしません．

さて，ここでPROLOG/Fの関数評価機能の説明を終わります．述語に関する使い方は普通のProlog処理系と同じなので，I/O '83年3月号などを参考にしてください．

エピローグ

現在，人工知能研究会では，Prologの記号処理能力とSmalltalkに見られるような対象を，自然に記述できる強力なシミュレーション機能を兼ね合わせたようなモデルを使って，推論するシステム〝Imagination Applier″というものを模索しております．

人工知能は未知なるアイディアの宝庫なのです．

◘参考文献

1) W.F. Clocksin/C.S. Mellish 中村克彦説:〝Prologプログラミング″，マイクロソフトウェア

2) 中島秀之:〝Prolog″，産業図書

リスト1 Prolog/FV.2 BASICプログラム・リスト

```
10 'SAVE"prologv2"
20 'SAVEM"PROLOGF2",&H1E00,&H564A,&H1E00
30 KEY1,CHR$(13)+";"+CHR$(13)+";"+CHR$(13)
40 KEY2,CHR$(13)+";"+CHR$(13)+";"+CHR$(13)+"pred "
50 KEY3,"exec"+CHR$(13)
60 KEY4,"func "
70 KEY5,"pred "
80 KEY6,"trace"+CHR$(13)
90 KEY10,"list"+CHR$(13)
100 WIDTH80,20:CONSOLE0,18,1:PRINT"Wait a minute!"
110 CLEAR300,&H1E00:DIM A$(4,4):LOADM"PROLOGF2"
```

リスト 1　Prolog/FV. 2 BASICプログラム・リスト

```
120 CONSOLE,20,0
130 ON ERROR GOTO 520
140 DEFUSR0=&H1E00:DEFUSR1=&H1E03:DEFUSR2=&H27D8
150 CLS:FORI=0 TO 1 STEP 3.14/30 :LINE(640,0)-(640*(1-COS(I)),50*SIN(I)),PSET,1:
NEXT
160 FORI=0 TO 3.14/2 STEP 3.14/20 :LINE(0,0)-(640*COS(I),80*SIN(I)),PSET,3:NEXT
170 SYMBOL(20,5)," Prolog with Functional calculation ",1,2,6,,NOT
180 SYMBOL(20,5)," Prolog with Functional calculation ",1,2,0,,PRESET
190 SYMBOL(10,20),"Prolog/F v2.0",6,7,5
200 SYMBOL(50,50)," The revolutionary language",2,2,2,,OR
210 SYMBOL(50,65),"     for the future generation",2,2,2,,OR:SYMBOL(320,80),"co
pyright (C) 1984 by E.Yoshikawa",1,1,4
220 LOCATE0,9:POKE&H1EF2,0:R%=USR(Q%)
230 ON R%+1 GOSUB 310,250,490
240 R%=USR1(Q%):GOTO230
250 LINEINPUT#1,A$
260 PRINT A$
270 IF ASC(A$)=&H23 THEN A$=MID$(A$,3)
280 Q%=USR2(INT(VARPTR(A$))):Q%=&H27F5:A$=""
290 IF EOF(1) THEN POKE&H1EF2,0:CLOSE#1
300 RETURN
310 PRINT"*^C is pressed*"
320 PRINT" Choose one!"
330 PRINT"   B for break"
340 PRINT"   C for cont"
350 PRINT"   I for input from file"
360 PRINT"   O for output to file"
370 INPUT"   P for output to printer ";A$
380 ON (INSTR("BbCcIiOoPp",A$)+1)¥2 GOTO 400,410,420,440,460
390 PRINT"???":GOTO320
400 STOP
410 RETURN
420 INPUT"File name please : ",F$:OPEN"I",#1,F$:POKE&H1EF2,&HFF:GOTO480
430 IF FO=1 THEN CLOSE#2:FO=0:GOTO470 ELSE 440
440 INPUT"File name please : ",F2$:OPEN"O",#2,F2$
450 POKE&H1F20,&HFF:FO=1:GOTO480
460 PO=1-PO
470 IF FO OR PO THEN POKE&H1F20,&HFF ELSE POKE&H1F20,0:POKE&H1F21,0
480 PRINT"OK.":GOTO320
490 IF FO THEN IF PEEK(&H1EF3)=10 THEN PRINT#2,LEFT$(OUT$,LEN(OUT$)-1):OUT$="" E
LSE OUT$=OUT$+CHR$(PEEK(&H1EF3))
500 IF PO THEN LPRINT CHR$(PEEK(&H1EF3));
510 RETURN
520 IF ERL=420 AND ERR=63 THEN PRINT"*File not found*":INPUT" Retry(Y-N) ";A$:IF
 A$="Y" OR A$="y" THEN RESUME 420 ELSE RESUME 480
530 IF ERL=440 AND ERR=64 THEN PRINT"*File already exist*" ELSE 620
540 PRINT" A for Another file name to input"
550 PRINT" E for Exit"
560 INPUT" K for Kill existing file ";A$
570 A=(INSTR("EeAaKk",A$)+1)¥2
580 IF A=0 THEN PRINT"???":GOTO550
590 IF A=1 THEN RESUME 440
600 IF A=2 THEN RESUME 480
610 IF A=3 THEN KILL F2$:RESUME
620 PRINT"ERROR IN LINE";ERL:ERROR ERR
```

リスト 2　Prolog/FV. 2 マシン語ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1E00 7E 20 00 BF 1E F0 1F 41 10 FE 1F 12 BF 1F 12 BE :B8
1E10 FF F6 10 BE 1F 14 10 BF FF F6 BF 1F 14 B7 FD 0D :6D
1E20 BD 20 41 35 40 BE 1E F0 AE 02 36 10 39 10 BE 1E :7A
1E30 F0 37 10 AF 22 34 40 1F 41 10 FE 1F 12 BF 1F 12 :0B
1E40 BE FF F6 10 BE 1F 14 10 BF FF F6 BF 1F 14 B6 FD :1D
1E50 0F 39 86 0A B7 1E A3 8E 1E 99 B6 FD 0F AD 9F FB :9E
1E60 FA B7 FD 0F B6 1E A5 F6 00 C3 3D FB 1E A4 89 00 :72
1E70 34 06 B6 1F 77 F6 00 C3 3D FB 1F 76 89 00 F3 1F :A7
1E80 70 10 A3 E1 26 03 7E 21 59 BE 1F 20 26 01 39 86 :08
1E90 0D BD 21 50 B6 0A 7E 21 50 11 00 1E A1 00 05 00 :8F
1EA0 05 00 00 00 01 13 7D 1E F2 26 03 7E 20 66 8E 00 :61
1EB0 01 36 10 BD 1E 2D 37 10 EC 01 FD 1F 6E A6 84 B7 :EE
1EC0 1F 71 7E 20 B2 34 36 8E 00 02 36 10 B7 1E F3 BD :A5
1ED0 1E 2D 37 10 35 B6 10 8E FC 00 1A 40 39 1C BF 10 :95
1EE0 83 E0 00 27 04 FD 1F 06 39 8E 27 7E 20 06 00 00 :42
1EF0 00 74 00 0A BD 21 67 7E 21 90 81 2C 81 20 1E FA :58
-------------------------------------------------------
Sum: 68 57 19 F8 B4 9C 65 76 F5 72 31 62 D9 77 DD 16 :38


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1F00 00 00 FF 00 00 00 D5 54 F0 00 00 00 00 00 00 00 :18
1F10 00 00 6F E6 21 85 FB F6 0A 00 00 00 00 00 00 00 :F6
1F20 00 00 00 0A 00 00 00 00 0D 11 07 23 20 11 07 00 :8A
1F30 00 00 00 00 00 00 00 00 0D 5D 5B 2E 2D 2C 29 28 :9D
1F40 20 00 00 00 00 00 00 00 0A 0D 20 00 00 00 00 00 :57
1F50 00 00 00 00 00 00 00 00 0A 5D 5B 2E 2D 2C 29 28 :9A
1F60 20 00 00 00 00 00 00 00 FF 00 00 00 12 00 1F 74 :C4
1F70 00 01 00 80 01 11 07 23 20 11 07 20 3B 65 78 28 :55
1F80 5B 27 20 70 72 69 6E 74 32 20 40 20 40 20 31 36 :48
1F90 20 62 61 73 65 20 21 20 70 72 5D 29 29 72 63 73 :F5
1FA0 2C 2A 70 72 70 5D 29 6E 67 65 28 5B 64 65 63 2C :43
1FB0 69 6E 63 5D 2C 5B 5D 29 29 29 29 5D 29 5D 5D 29 :88
1FC0 2C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :2C
1FD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1FE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1FF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-------------------------------------------------------
Sum: 7C 22 C2 22 95 D7 EC 98 79 09 D2 A0 BD 22 44 EA :73


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2000 BF 1E F0 BE FF F6 BF 1F 14 7E 21 AA 7D 1F 71 26 :EE
2010 15 7D 1F 68 26 08 86 FF B7 1F 68 86 0D 39 86 FF :5B
2020 B7 1F 71 86 0A 39 34 76 7D 1F 68 27 05 86 13 BD :40
2030 1E A6 BE 1F 6E A6 80 A7 E4 BF 1F 6E 7A 1F 71 35 :4B
2040 F6 B6 FD 0F BD DA F6 B7 FD 0F 39 6E 8E 00 04 BF :00
2050 1F 70 8E 0C 17 BF 1F 76 8E 1F 6C B6 FD 0F AD 9F :BB
2060 FB FA B7 FD 0F 39 B7 1F 6C 8E 1F 74 BF 1F 6E 8E :2E
2070 1F 28 10 8E 1F 74 5F A6 80 27 05 A7 A0 5C 20 F7 :E3
2080 1D FD 1F 70 8E 00 80 BF 1F 72 8E 1F 6C B6 FD 0F :E2
2090 AD 9F FB FA B7 FD 0F B6 1F 74 81 02 27 4D 81 01 :C6
20A0 27 4E B6 1F 71 80 04 23 42 B7 1F 71 8E 1F 78 BF :CF
20B0 1F 6E 7F 1F 68 BE 1F 20 27 28 8E 1F 28 A6 80 81 :5B
20C0 01 23 0D 81 20 25 F4 12 12 12 12 BD 21 50 20 ED :6E
20D0 BE 1F 6E F6 1F 71 A6 80 BD 1E C5 5A 26 F8 12 12 :33
20E0 12 12 8E 00 00 BF 1F 26 7E 1E 52 86 12 7E 1E A6 :7E
20F0 8E 00 00 36 10 BD 1E 2D BD 20 41 7E 20 EB 12 B6 :4B
-------------------------------------------------------
Sum: 47 54 E8 C6 0C 70 AD CA 54 91 FF D0 B5 00 92 A5 :DC


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2100 FD 0F 39 BD 20 41 20 E3 34 14 20 17 34 14 BE 1F :0A
2110 26 27 03 7E 21 90 BE 1F 1E 27 08 BE 1F 02 A7 80 :AF
2120 BF 1F 02 C6 14 BD 21 34 BE 1F 20 27 05 C6 0E BD :86
2130 1E C5 35 94 34 72 1F 98 30 E4 10 8E 00 01 34 36 :26
2140 30 E4 B6 FD 0F AD 9F FB FA B7 FD 0F 32 66 35 F2 :99
```

リスト 2　Prolog/FV. 2 マシン語ダンプ・リスト

```
2150 7E 1E C5 0E BD 21 34 35 84 34 02 86 0D BD 21 0C :ED
2160 86 0A BD 21 0C 35 82 34 02 A6 80 81 04 27 05 BD :FB
2170 21 0C 20 F5 35 82 BD 20 0C 34 04 F6 1F 18 B7 1F :1D
2180 18 1F 98 35 84 34 10 7E 22 A8 BF 1F 26 35 10 3B :98
2190 86 FF B7 1F 68 B7 1F 71 8E 00 00 FE 1F 16 BF 1F :A9
21A0 1E BF 1F 26 10 CE 70 00 20 5B B7 FD 0F 10 FF 1F :DC
21B0 12 10 CE 70 00 7F 1F 02 12 CE FC 00 BD 25 05 8E :51
21C0 E0 00 AF 1E 30 1C 8C 90 00 26 F7 BF 1F 06 8E F0 :94
21D0 00 BF 1F 08 8E 1E FA BF 1E FE 8E 81 20 BF 1E FC :6F
21E0 8E 81 2C BF 1E FA 8E 80 10 34 10 34 10 8E 00 00 :46
21F0 BF 1F 1C BF 1F 20 BF 1F 26 8E 21 85 BF FF F6 8E :72
--------------------------------------------------------
Sum: 50 7E 1D 44 8D 11 C1 31 02 BA 03 A9 D9 11 2E ED :2C


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2200 00 0A BF 1F 22 8E 00 00 BF 1F 24 BD 21 59 B7 FD :85
2210 0F 86 00 B7 1F 04 8E 0D 11 BF 1F 28 8E 07 23 BF :98
2220 1F 2A 8E 20 11 BF 1F 2C 8E 07 00 BF 1F 2E 8E 0D :4E
2230 20 BF 1F 38 8E 00 00 BF 1F 3A 8E 20 0D BF 1F 48 :BD
2240 8E 0A 00 BF 1F 4A 8E 00 00 BF 1F 58 8E 00 00 BF :D1
2250 1F 1E BD 20 41 7D 1F 02 26 0F 73 1F 02 8E 00 00 :50
2260 BF 1F 1E BD 3F 07 FF 1F 16 86 12 BD 20 66 BD 21 :EC
2270 76 12 12 12 12 7E 3F B8 50 52 4F 4C 4F 47 2F 46 :7B
2280 20 56 32 2E 30 20 43 6F 70 79 20 72 69 67 68 74 :FF
2290 20 28 43 29 20 62 79 20 45 69 69 63 68 69 20 59 :93
22A0 6F 73 68 69 6B 61 77 61 AE 63 8C 80 00 25 09 8E :30
22B0 FF FF BF 1F 26 35 10 3B 8E 21 90 AF 63 35 10 3B :53
22C0 AE C4 8C 80 10 26 01 39 7E 28 59 01 26 0E 36 10 :68
22D0 8E 22 A1 BD 21 67 BD 25 53 7E 21 90 84 08 27 02 :AF
22E0 8D 04 AD 94 20 8B 34 10 BD 23 8B 36 10 35 10 39 :F0
22F0 A6 02 84 02 26 90 36 10 16 FF 76 10 BE 1F 1A 26 :E2
--------------------------------------------------------
Sum: 4D AE 53 8E E9 5D 03 7A 9E F3 E4 1F 86 1C 9B 3E :AE


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2300 74 A6 02 85 10 26 4E 85 04 26 62 10 8E 1E 7E BD :2D
2310 24 ED 26 18 10 BE 1F 08 C6 AE E7 A0 8B 60 A7 A0 :71
2320 8E 36 10 AF A1 10 BF 1F 08 16 FF 45 10 BE 1F 08 :69
2330 EC 84 10 83 F0 00 24 0C 10 83 25 16 27 06 1F 01 :3E
2340 86 BD 20 06 86 AD A7 A0 86 9F A7 A0 AF A1 10 BF :6E
2350 1F 08 16 FF 1C 85 08 27 02 8D 8B A6 02 84 01 27 :7A
2360 07 AE 84 AD 02 16 FF 09 AD 94 16 FF 04 A6 02 84 :8C
2370 20 10 26 FF 53 10 BE 1F 08 86 8E A7 A0 AF A1 8E :D6
2380 36 10 AF A1 10 BF 1F 08 16 FE E6 10 BE 1F 1C 27 :B6
2390 2D 10 BE 63 FC 10 8C 80 10 26 0A 10 8E 00 00 10 :64
23A0 BF 1F 1C 20 19 AE A4 10 AE 22 10 BF 63 FC 10 8E :31
23B0 FF FF 8C 80 00 25 02 31 21 10 BF 1F 1A 39 C6 00 :8A
23C0 BE 1F 1E 27 04 8D 25 20 03 BD 21 76 8E 1F 48 BD :01
23D0 24 0D 26 EC 34 02 5C 8E 1F 58 BD 24 0D 26 38 B6 :DC
23E0 1F 18 8E 1F 38 BD 24 0D 26 2D 20 D4 34 04 BE 1F :66
23F0 02 E6 80 5D 26 0B 8E 00 00 BF 1F 1E 35 04 7E 21 :58
--------------------------------------------------------
Sum: 02 38 8F B3 63 45 40 2B 5C 0A 1F 81 72 5D C5 D6 :FF


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2400 76 BF 1F 02 1F 98 34 02 BD 21 08 35 86 6D 84 27 :FC
2410 05 A1 80 26 F8 4D 39 5A A6 E5 5C 81 24 27 50 81 :A8
2420 30 25 64 81 3A 24 60 8D 03 32 E5 39 36 04 8E 00 :A0
2430 00 36 10 5C 5C BE 1F 22 36 10 34 04 BD 28 CB 35 :60
2440 04 5A A6 E5 81 60 25 02 80 20 80 30 81 0A 25 02 :F3
2450 80 07 34 04 1F 89 4F 36 06 BD 28 BD 35 04 C1 02 :90
2460 26 D3 37 10 37 04 10 8E FF FF 10 BF 1F 1A 39 BE :16
2470 1F 22 36 10 8E 00 10 BF 1F 22 5A 8D AF 5C 37 20 :6E
2480 10 BF 1F 22 32 E5 39 8E 80 00 6D 02 27 34 A6 03 :E1
2490 31 86 31 24 36 22 E1 C0 26 10 31 E6 30 04 A6 A2 :CE
24A0 A1 80 27 FA 30 1F AC C4 24 04 37 10 20 DC 37 10 :B3
24B0 1F 98 40 30 86 30 1C 32 E5 10 8E 00 00 10 BF 1F :9C
24C0 1A 39 10 8E 25 16 10 AF 84 86 01 A7 02 E7 03 36 :BF
24D0 14 30 04 31 E5 A6 A2 A7 80 5A 26 F9 86 00 A7 02 :75
24E0 37 14 32 E5 10 8E 00 00 10 BF 1F 1A 39 34 20 10 :A5
24F0 AE A4 4F 10 AC E4 27 08 AC A1 27 06 8B 02 20 F3 :8A
--------------------------------------------------------
Sum: 88 8F A6 32 F6 38 3B 32 AF AA 5F E4 E4 85 AF CE :0C


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2500 86 FF 35 20 39 8E 4D DE 10 8E 80 00 A6 80 A7 A0 :57
2510 8C 56 4B 25 F7 39 34 10 8E 25 3A BD 21 67 35 10 :3D
2520 10 AE E4 10 AE 3E 10 BC 25 38 12 12 36 10 20 02 :53
2530 36 20 BD 25 53 7E 21 90 AD 94 55 6E 64 65 66 69 :56
2540 6E 65 64 20 57 6F 72 64 20 45 72 72 6F 72 20 2D :6A
2550 3E 20 04 8E 7F FF AC C4 25 03 7E 29 55 37 10 8C :D5
2560 90 00 24 0D E6 03 30 04 A6 80 BD 21 0C 5A 26 F8 :66
2570 39 86 28 BD 21 0C 36 10 AE 84 36 10 8D D5 37 10 :38
2580 AE 02 8C 90 00 25 07 86 20 BD 21 0C 20 E8 8C 80 :9C
2590 10 27 0A 86 2E BD 21 0C 36 10 BD 25 53 86 29 BD :C6
25A0 21 0C 39 BD 25 53 BE 1F 24 26 03 7E 21 59 39 20 :16
25B0 00 B6 1F 04 B7 1F 0E 8B 01 B7 1F 04 39 B6 1F 0E :3F
25C0 B7 1F 04 39 10 BE 1F 06 10 8C E0 00 27 1E AE 22 :97
25D0 BF 1F 06 AE 42 AF A4 37 10 AF 22 1F 20 44 56 44 :5C
25E0 56 C3 BC 00 1F 01 6F 84 10 AF C4 39 30 C4 BD 1E :73
25F0 D6 12 8D 0C 30 E4 10 8E 70 00 8D 04 8D 6D 20 C4 :12
--------------------------------------------------------
Sum: 4E 2C 16 BC B9 A6 6C 01 24 5F 57 18 8F 44 DD 8F :49
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2600 34 30 10 8E 00 00 AE 84 8D 13 35 10 30 02 AC E4 :DB
2610 35 20 25 EC 39 6C 21 EC 22 AF 22 1F 01 8C 90 00 :47
2620 25 3B 8C E0 00 24 36 1F 10 44 56 25 30 44 56 25 :03
2630 2C C3 BC 00 1E 01 6D 84 2A 04 1E 01 20 1F 63 84 :2E
2640 1F 01 EC 84 10 AF 84 1F 12 1F 01 20 D0 EC A4 C4 :68
2650 FE 34 06 EC 22 ED A4 AF 22 1F 21 35 20 10 8C 00 :D9
2660 00 26 01 39 E6 21 54 24 AC 20 E2 8E E0 00 1F 10 :2A
2670 10 8E F4 00 30 1C 6D A2 2A 04 63 A4 20 04 ED 02 :35
2680 1F 10 8C 90 00 26 ED BD 1E DD 39 AE C4 36 10 39 :40
2690 20 02 20 23 37 10 10 8E 1E FE BD 24 ED 27 11 10 :7C
26A0 BE 1E FE AF A3 10 BF 1E FE 10 AE E4 34 20 86 00 :93
26B0 37 10 8B 02 AF E6 39 37 10 10 8E 1E 7E BD 24 ED :F1
26C0 27 1D 10 BE 1E 7E AF A3 10 BF 1E 7E 10 BE 1F 08 :60
26D0 8E 37 10 AF A1 8E 34 10 AF A1 10 BF 1F 08 39 10 :86
26E0 BE 1F 08 8E 37 10 AF A1 C6 AF E7 A0 8B 60 A7 A0 :38
26F0 10 BF 1F 08 39 37 10 10 BE 1F 08 10 BF 1F 0A 31 :94
--------------------------------------------------------
Sum: 9E A9 E0 6A 57 E9 F2 AB 80 95 81 9D 4D 70 05 82 :E5


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2700 21 10 AF 84 10 BF 1F 08 BF 1F 0C 86 02 B7 1F 04 :A6
2710 8E 1E 7E BF 1E 7E 8E 1D FE BF 1D FE 39 10 BE 1D :2C
2720 FE 10 8C 1D FE 26 2A 86 00 B7 1F 04 10 BE 1F 08 :5A
2730 CC 1E 7E 1F 01 A3 84 27 06 4F C3 32 60 ED A1 86 :94
2740 39 A7 A0 10 BF 1F 08 1F 20 B3 1F 0A E7 9F 1F 0A :40
2750 39 86 00 B7 1F 04 AE A1 36 10 BD 25 53 8E 27 7D :95
2760 BD 21 67 AE A4 36 10 BD 25 53 BE 1F 0A BF 1F 08 :DF
2770 BE 1F 0C 10 8E 25 16 10 AF 84 7E 21 90 20 0D 0A :6B
2780 4F 76 65 72 66 6C 6F 77 21 04 8E 80 44 AC C1 26 :5E
2790 25 8E 27 BE BD 21 67 BD 3C 04 BD 28 54 BD 28 54 :4C
27A0 BD 3E BD 8E 27 C7 BD 21 67 BD 3C 11 BD 28 54 BD :79
27B0 28 54 BD 3E BD 39 8E 84 8A 36 10 7E 25 A3 2A 70 :2F
27C0 72 65 64 2A 0D 0A 04 0D 0A 2A 66 75 6E 63 2A 0D :A4
27D0 0A 04 2E 0D 0A 0D 0A 04 AE 02 E6 84 AE 01 10 8E :D5
27E0 1F 74 34 24 A6 80 A7 A0 5A 26 F9 35 24 F7 27 F5 :3D
27F0 10 BF 27 F6 39 FE 16 FF 24 8E 28 02 BD 21 67 7E :D7
--------------------------------------------------------
Sum: 6A FB 3D 51 3A A6 23 E8 71 59 27 90 F6 2E 3E FD :BE


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2800 27 6A 54 6F 6F 4C 6F 6E 67 04 BE 1D FE 8C 1D FE :D7
2810 27 1E 10 AE 84 10 8C 80 36 26 15 FC 1F 08 A3 04 :DE
2820 10 83 00 80 24 D3 5A E7 98 04 30 06 BF 1D FE 39 :30
2830 10 8E 80 36 10 AF 83 10 8E 80 44 10 AF 83 BF 1D :16
2840 FE 16 FE D9 8E FF FF 37 06 10 83 80 10 27 02 30 :30
2850 01 36 10 39 AE D4 AF C4 39 AE C4 AE 02 AF C4 39 :7C
2860 20 00 8E 80 10 36 10 B6 1F 04 B7 1F 0E 86 01 B7 :7F
2870 1F 04 39 B6 1F 0E 27 14 AE C4 8C 80 10 27 05 17 :4B
2880 FE 35 20 F4 33 42 86 02 B7 1F 04 39 86 00 B7 1F :B3
2890 04 8E 80 10 36 10 AE 42 8C 80 10 27 05 BD 25 C4 :46
28A0 20 F4 37 10 AF C4 39 37 06 1F 98 BD 21 0C BE 1F :C2
28B0 24 26 03 BD 21 59 39 BD 23 8B 36 10 39 37 06 E3 :C7
28C0 C4 ED C4 39 EC 42 A3 C1 ED C4 39 33 5C 8E 00 00 :47
28D0 AF C4 A6 45 E6 47 3D ED 42 A6 44 E6 47 3D E3 41 :6F
28E0 ED 41 24 02 6C C4 EC 45 3D E3 41 ED 41 24 02 6C :D6
28F0 C4 A6 44 E6 46 3D E3 C4 ED C4 AE 42 AF 46 33 46 :CD
--------------------------------------------------------
Sum: 16 5E 65 52 4F EE 12 99 94 8E 1F 71 33 EC 01 67 :4C


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2900 39 BD 29 0A BD 29 3C 7E 29 39 8E 00 00 36 10 86 :85
2910 10 34 02 68 45 69 44 69 41 69 C4 EC C4 10 A3 42 :1C
2920 25 0B A3 42 ED C4 EC 44 C3 00 01 ED 44 6A E4 26 :5F
2930 E2 32 61 8D 0E 8D 0C 20 00 37 10 39 37 30 1E 12 :E0
2940 36 30 39 37 36 36 16 36 20 39 8D 09 BE 1F 24 26 :A4
2950 03 7E 21 59 39 4F AE C4 26 07 33 42 86 30 7E 21 :EC
2960 0C 34 02 BE 1F 22 36 10 8D A0 8D D0 37 06 35 02 :85
2970 34 04 4C AE C4 27 02 20 E8 33 42 35 04 CB 30 C1 :91
2980 3A 25 02 CB 07 1E 89 BD 21 0C 1F 98 4A 26 EC 39 :10
2990 BE 1F 22 34 10 8E 00 10 BF 1F 22 8D AD 35 10 BF :1F
29A0 1F 22 39 8E 81 35 36 10 39 8E 1F 22 36 10 39 8E :19
29B0 1F 28 36 10 39 8E 1F 48 36 10 39 8E 1F 38 36 10 :65
29C0 39 8E 1F 58 36 10 39 20 02 20 23 37 10 10 8E 1E :25
29D0 FE BD 24 ED 27 11 10 BE 1E FE AF A3 10 BF 1E FE :2B
29E0 10 AE E4 34 20 86 00 8B 02 30 E6 36 10 39 37 10 :E5
29F0 10 8E 1E 7E BD 24 ED 27 22 10 BE 1E 7E AF A3 10 :1D
--------------------------------------------------------
Sum: 56 29 AF D1 5A EB 88 2A 7B 13 01 65 B8 5A AD DC :85


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2A00 BF 1E 7E 10 BE 1F 08 8E 30 7E AF A1 8E 34 10 AF :5D
2A10 A1 8E 36 10 AF A1 10 BF 1F 08 39 10 BE 1F 08 C6 :AF
2A20 30 E7 A0 8B 60 A7 A0 8E 36 10 AF A1 10 BF 1F 08 :03
2A30 39 E6 D4 4F ED C4 39 37 30 10 AF 84 39 37 16 1E :7A
2A40 01 E7 84 39 8E 81 3B 36 10 39 8E 1F 1C 36 10 39 :B6
2A50 10 8E FF FF AE C1 27 02 31 21 36 20 39 37 10 10 :6C
2A60 8E FF FF AC C4 27 02 31 21 10 AF C4 39 BE 1D FE :0C
2A70 10 BE 1F 08 10 AF 83 10 8E 81 56 10 AF 83 10 8E :8C
2A80 81 4C 10 AF 83 BF 1D FE 39 BE 1D FE 10 AE 84 10 :4D
2A90 8C 81 4C 27 14 10 8E 81 4C 10 AF 83 10 8E 81 56 :B6
2AA0 10 AF 83 BF 1D FE 16 FC 74 10 BE 1F 08 CC AE C1 :D2
2AB0 ED A1 86 27 A7 A1 34 20 EC 04 A3 E1 10 83 FF 80 :5D
2AC0 25 0C E7 3F 10 BF 1F 08 30 06 BF 1D FE 39 83 00 :19
2AD0 02 ED A1 CC 10 27 ED 3C 20 EA 8E 1F 20 36 10 39 :12
2AE0 8E 1F 1A 36 10 39 BE 1D FE 10 AE 84 10 8C 81 4C :CA
```

リスト 2　Prolog/FV. 2 マシン語ダンプ・リスト

```
2AF0 27 14 10 8E 81 4C 10 AF 83 10 8E 81 74 10 AF 83 :BD
-------------------------------------------------------
Sum: 5E F4 E0 71 D6 1C A7 36 5B 83 C5 AB AC 8D 0F 1F :27


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2B00 BF 1D FE 7E 27 1D 10 BE 1F 08 CC AE C1 ED A1 C6 :20
2B10 27 E7 A0 10 AF 83 31 21 10 BF 1F 08 10 8E 81 7D :D4
2B20 10 AF 83 10 8E 81 74 10 AF 83 BF 1D FE 39 BE 1D :05
2B30 FE 10 AE 84 10 8C 81 74 27 14 10 8E 81 74 10 AF :5E
2B40 83 10 8E 81 7D 10 AF 83 BF 1D FE 7E 27 1D 10 BE :CB
2B50 1F 08 C6 16 E7 A4 31 23 34 20 EC 0A A3 E1 ED 3E :DB
2B60 10 BF 1F 08 31 3F 1F 20 A3 04 10 83 00 80 10 24 :93
2B70 FC 87 E7 98 04 30 0C BF 1D FE 39 AE 42 36 10 39 :C4
2B80 8E 25 06 36 10 39 8E 25 11 36 10 39 8E 1F 02 36 :60
2B90 10 39 37 06 34 04 37 10 34 10 37 10 34 10 37 10 :1B
2BA0 34 10 37 10 34 10 37 06 34 04 37 06 34 04 86 15 :54
2BB0 34 02 31 7E 8E 00 0E 34 10 34 20 8E 10 00 34 10 :FB
2BC0 30 E4 B6 FD 0F AD 9F FB FA B7 FD 0F 32 E8 12 39 :3F
2BD0 BE 1F 08 36 10 39 20 00 37 06 F7 1F 04 39 FC 1F :2F
2BE0 08 E3 C1 FD 1F 08 39 BE 1F 08 37 06 ED 81 BF 1F :77
2BF0 08 39 BE 1F 08 37 06 E7 80 BF 1F 08 39 39 02 37 :5B
-------------------------------------------------------
Sum: A6 B0 0B 72 59 42 49 F7 11 9F D5 33 BE EA CF 81 :5E


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2C00 10 32 62 7E 23 75 75 8E 22 65 36 10 39 8E 25 16 :8C
2C10 36 10 39 10 8E FF FF AE C1 27 04 AE C4 26 02 31 :80
2C20 21 10 AF C4 39 37 06 AA C4 EA 41 ED C4 39 10 8E :3B
2C30 FF FF 37 06 10 A3 C4 22 02 31 21 10 AF C4 39 10 :F4
2C40 8E FF FF 37 06 10 A3 C4 25 02 31 21 10 AF C4 39 :75
2C50 BD 23 8B 10 BE 1F 08 10 BF 1F 0A BF 1F 0C 31 21 :94
2C60 10 AF 84 8E 30 8C AF A1 86 08 A7 A0 8E 36 10 AF :35
2C70 A1 86 BE A7 A0 10 BF 1F 10 31 22 8E 6E 88 AF A1 :51
2C80 31 21 10 BF 1F 08 39 20 31 BD 27 1D 10 BE 1F 08 :C8
2C90 31 3F 86 8E A7 A0 BE 1F 0C AF A1 86 BD A7 A0 8E :1C
2CA0 2C 87 AF A1 10 BF 1F 08 86 02 B7 1F 04 8E 1E 7E :85
2CB0 BF 1E 7E 8E 1D FE BF 1D FE 39 10 BE 1F 10 AF A4 :67
2CC0 35 06 A3 84 E7 24 FC 1F 08 B3 1F 0A E7 9F 1F 0A :1B
2CD0 39 BD 25 C4 6C 84 39 10 8E FF FF 37 06 44 56 44 :BF
2CE0 56 C3 BC 00 1F 01 A6 84 44 25 02 31 21 36 20 39 :6B
2CF0 8E 13 06 20 00 6F E2 10 8E 00 FF CC 28 19 34 36 :2C
-------------------------------------------------------
Sum: 01 46 9A B8 F3 96 E9 C3 4C 7F 4E 87 C1 5F 79 04 :0B


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2D00 CC 01 00 8E 00 0B 31 7C 1E 20 34 36 86 10 34 06 :8B
2D10 30 E4 B6 FD 0F AD 9F FB FA B7 FD 0F 32 6F 39 8E :42
2D20 00 19 20 D1 37 10 34 10 37 10 34 10 37 10 34 10 :AB
2D30 AE 60 36 10 BD 28 54 BD 2A 31 AE 62 36 10 BD 28 :E0
2D40 54 BD 2A 3D AE 60 36 10 BD 28 54 8E 00 01 36 10 :DA
2D50 BD 28 BD AE 60 36 10 BD 2A 37 AE 62 36 10 BD 28 :4F
2D60 54 8E 00 01 36 10 BD 28 BD AE 62 36 10 BD 2A 37 :3F
2D70 AE 64 36 10 8E 00 01 36 10 BD 28 C4 37 10 AF 64 :30
2D80 AE 64 36 10 BD 2A 50 AE C1 27 A5 32 66 39 BD 28 :80
2D90 B7 37 10 34 10 AE 60 36 10 BD 28 54 37 10 34 10 :5A
2DA0 BD 2B 80 BD 28 54 37 10 34 10 AE 62 36 10 8E 00 :10
2DB0 01 36 10 BD 28 C4 BD 2A 31 8E 00 01 36 10 BD 28 :C2
2DC0 C4 37 10 34 10 30 64 36 10 30 62 36 10 AE 60 36 :45
2DD0 10 AD 9F 82 47 BD 2B 80 BD 28 54 AE 66 36 10 BD :DD
2DE0 2A 37 AE 62 36 10 BD 2B 80 BD 2A 37 32 68 39 8E :9E
2DF0 80 00 36 10 37 10 34 10 BD 2B 80 BD 28 54 37 10 :39
-------------------------------------------------------
Sum: 5E 4C 92 4E B6 93 80 7E 6D A4 7A 62 7B 86 46 90 :95


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2E00 34 10 30 62 36 10 30 60 36 10 8E 00 03 36 10 BD :86
2E10 2D 24 AE 62 36 10 8E 00 01 36 10 BD 28 C4 BD 2A :0C
2E20 31 AE C1 27 1D 30 62 36 10 30 60 36 10 AE 62 36 :D8
2E30 10 BD 2A 31 8E 00 01 36 10 BD 28 BD BD 2D 24 16 :C3
2E40 FF C0 AE 60 36 10 BD 2B 86 BD 2A 37 AE 60 36 10 :F3
2E50 8E 00 01 36 10 BD 28 C4 BD 29 90 32 64 39 37 10 :0A
2E60 34 10 AE 60 36 10 BD 28 54 37 10 34 10 8E 00 01 :EB
2E70 36 10 BD 29 A9 8E 00 02 36 10 BD 28 BD BD 2A 37 :6B
2E80 AE 60 36 10 BD 28 54 BD 2C 0D BD 2A 5D AE C1 27 :5D
2E90 0A 8E 82 94 36 10 BD 25 A3 20 12 8E 00 24 36 10 :A3
2EA0 BD 28 A7 AE 60 36 10 BD 28 54 BD 29 90 BD 2A 44 :BA
2EB0 BD 25 A3 8E 00 00 36 10 BD 29 A9 8E 00 02 36 10 :BE
2EC0 BD 28 BD BD 2A 37 AE 60 36 10 BD 25 A3 AE 60 36 :DD
2ED0 10 8E 00 03 36 10 BD 28 BD BD 2A 31 8E 00 04 36 :69
2EE0 10 BD 28 BD AE 60 36 10 BD 28 BD AE 62 36 10 BD :BB
2EF0 2A 37 32 64 39 8E 80 00 36 10 37 10 34 10 AE 60 :1D
-------------------------------------------------------
Sum: D2 64 FC FC D6 5E 3B 2C BE 0F BD F8 8B 3E 63 9F :16


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2F00 36 10 8E 00 02 36 10 BD 28 BD BD 2A 31 AE C1 27 :6C
2F10 0A 30 60 36 10 BD 2E 5E 16 FF E3 32 62 39 37 10 :35
2F20 34 10 8E 00 02 36 10 37 10 34 10 30 62 36 10 37 :B4
2F30 10 34 10 8E 14 00 36 10 37 10 34 10 30 E4 B6 FD :8E
2F40 0F AD 9F FB FA B7 FD 0F 32 68 39 37 10 34 10 37 :A8
2F50 10 34 10 AE 62 36 10 BD 28 59 BD 28 44 AE C1 27 :A7
2F60 1D AE 60 36 10 8E 80 10 36 10 BD 25 C4 AE 62 36 :C1
2F70 10 8E 00 02 36 10 BD 28 BD BD 2A 37 20 64 AE 60 :38
2F80 36 10 AE 62 36 10 AD 9F 82 BD BD 2A 5D AE C1 27 :01
2F90 10 AE 60 36 10 AE 62 36 10 BD 28 59 BD 2A 37 20 :36
2FA0 41 AE 60 36 10 AE 62 36 10 AD 9F 82 BD BD 2C 2E :8D
2FB0 AE C1 27 1F AE 60 36 10 AE 62 36 10 BD 28 59 BD :5A
2FC0 25 C4 AE 62 36 10 8E 00 02 36 10 BD 28 BD BD 2A :9E
2FD0 37 20 0F AE 60 36 10 AE 62 36 10 BD 28 59 AD 9F :9A
2FE0 82 D3 32 64 39 BD 28 59 BD 28 54 BD 28 54 BD 29 :BA
2FF0 3C BD 28 54 BD 29 3C 39 37 10 34 10 37 10 34 10 :E6
-------------------------------------------------------
Sum: 1F 42 47 5A 5A AC 77 C1 7A BB 23 B3 A0 2C 77 93 :21


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3000 AE 60 36 10 AE 62 36 10 BD 2A 3D AE 60 36 10 8E :B0
3010 00 00 36 10 BD 2A 5D AE C1 27 02 20 10 AE 62 36 :98
3020 10 8E 00 01 36 10 BD 28 BD AD 9F 82 F2 32 64 39 :16
3030 37 10 34 10 AE 60 36 10 BD 29 A9 BD 28 54 BD 29 :8D
3040 0A 37 10 AF 60 8E 00 30 36 10 BD 28 BD BD 26 8B :74
3050 8E 00 39 36 10 BD 2C 3F AE C1 27 08 8E 00 07 36 :9E
3060 10 BD 28 BD AE 60 36 10 BD 2A 50 AE C1 27 C5 32 :CA
3070 62 39 37 10 34 10 37 10 34 10 AE 62 36 10 BD 28 :EC
3080 59 BD 28 44 AE C1 27 02 20 42 AE 60 36 10 AE 62 :E0
3090 36 10 BD 28 59 BD 28 54 BD 28 54 BD 2A 5D AE C1 :A9
30A0 27 1B AE 62 36 10 BD 28 59 BD 28 59 AE 62 36 10 :6A
30B0 8E 00 02 36 10 BD 28 BD BD 2A 37 20 0F AE 60 36 :09
30C0 10 AE 62 36 10 BD 28 59 AD 9F 83 0C 32 64 39 37 :85
30D0 10 34 10 AE 60 36 10 BD 28 54 BD 28 44 AE C1 27 :A0
30E0 14 8E 80 10 36 10 8E 80 10 36 10 BD 25 C4 AE 60 :90
30F0 36 10 BD 2A 37 BD 28 B7 BD 26 8B AE 60 36 10 BD :7F
-------------------------------------------------------
Sum: AD 93 8C 05 CB C2 41 0D 62 D2 A5 82 E4 E7 EC 25 :E3


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3100 28 54 AD 9F 83 20 BD 29 3C 8E 00 00 36 10 BD 29 :47
3110 B5 BD 2F F8 8E 00 00 36 10 8E 00 20 36 10 8E 00 :EF
3120 0A 36 10 8E 00 28 36 10 8E 00 29 36 10 8E 00 5B :32
3130 36 10 8E 00 5D 36 10 8E 00 7C 36 10 BD 29 C1 BD :2B
3140 2F F8 8E 00 00 36 10 8E 00 20 36 10 8E 00 0D 36 :C0
3150 10 8E 00 28 36 10 8E 00 29 36 10 8E 00 5B 36 10 :38
3160 8E 00 5D 36 10 8E 00 7C 36 10 BD 29 BB BD 2F F8 :06
3170 BD 2B 7B AD 9F 83 32 8E 00 00 36 10 BD 29 3C 8E :E8
3180 00 20 36 10 BD 29 3C BD 30 30 8E 00 C5 36 10 8E :CC
3190 00 11 36 10 8E 00 01 36 10 BD 34 98 BD 2F F8 AD :46
31A0 9F 83 3D BD 25 C4 BD 2B 7B BD 2A 37 BD 26 8B AD :A1
31B0 9F 83 4A 8E 00 00 36 10 8E 00 20 36 10 8E 00 0D :CF
31C0 36 10 8E 00 0A 36 10 BD 29 B5 BD 2F F8 32 62 39 :70
31D0 37 10 34 10 37 10 34 10 AE 62 36 10 BD 28 59 BD :67
31E0 28 44 AE C1 27 29 8E 80 10 36 10 8E 80 10 36 10 :F3
31F0 BD 25 C4 BD 26 8B 8E 80 10 36 10 BD 25 C4 AE 62 :2E
-------------------------------------------------------
Sum: 37 C8 07 29 51 BC 63 90 79 2B B7 CC E8 5F EC 6A :F3


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3200 36 10 8E 00 02 36 10 BD 28 BD BD 2A 37 20 32 AE :DC
3210 62 36 10 BD 28 59 BD 28 54 BD 26 8B BD 28 54 BD :83
3220 28 54 AE 60 36 10 BD 2A 5D AE C1 27 02 20 12 BD :9B
3230 29 39 AE 60 36 10 AE 62 36 10 BD 28 59 AD 9F 83 :19
3240 20 32 64 39 37 10 34 10 8E 00 0A 36 10 BD 29 A9 :E7
3250 BD 2A 37 8E 00 01 36 10 BD 29 A9 8E 00 02 36 10 :58
3260 BD 28 BD BD 2A 37 8E 00 64 36 10 AE 60 36 10 BD :09
3270 28 59 37 10 AF 60 AE 60 36 10 BD 28 44 BD 2A 50 :8B
3280 AE C1 27 46 BD 26 8B AE 60 36 10 BD 28 54 BD 2A :BE
3290 37 8E 00 0A 36 10 BD 28 BD 8E 11 85 36 10 BD 2F :0D
32A0 1E AE 60 36 10 BD 28 54 BD 26 8B BD 28 54 BD 25 :34
32B0 A3 BD 28 59 BD 26 8B BD 28 44 AE C1 27 EA BD 29 :DE
32C0 39 BD 29 A3 BD 25 A3 16 FF A1 8E 00 00 36 10 BD :8E
32D0 29 A9 8E 00 02 36 10 BD 28 BD BD 2A 37 32 62 39 :35
32E0 BD 28 B7 BD 26 8B BD 29 A3 BD 2A 5D AE C1 27 0A :77
32F0 BD 29 39 8E 80 10 36 10 20 53 BD 26 8B 8E 80 31 :A3
-------------------------------------------------------
Sum: 2D 21 DF DE CB 66 7F E4 E0 43 6D 0B 20 20 DD 49 :A0


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3300 36 10 BD 2A 5D AE C1 27 02 20 42 AD 9F 83 3D BD :4D
3310 26 8B 8E 80 31 36 10 BD 2A 5D AE C1 27 08 BD 29 :FE
3320 3C BD 29 39 20 27 BD 26 8B BD 28 44 AE C1 27 1A :E9
3330 BD 2B 7B BD 2A 44 BD 2A 5D AE C1 27 08 BD 29 3C :92
3340 BD 29 39 20 03 BD 25 C4 20 03 BD 25 C4 39 37 10 :31
3350 34 10 8E 00 00 36 10 37 10 34 10 AD 9F 83 5D BD :8C
3360 26 8B 8E 80 31 36 10 BD 2A 5D BD 2A 50 AE C1 27 :47
3370 67 BD 26 8B 8E 7F FF 36 10 BD 2C 3F AE C1 27 0F :F4
3380 BD 32 E0 BD 25 C4 AE 62 36 10 BD 2A 37 20 46 AE :FD
3390 60 36 10 AE C1 27 02 20 30 8E 00 00 36 10 BD 2B :4A
33A0 7B 8E 00 20 36 10 BD 29 3C 8E 00 0A 36 10 BD 28 :54
33B0 BD BD 30 30 8E 00 05 36 10 8E 00 11 36 10 8E 00 :26
33C0 01 36 10 BD 34 98 BD 2F F8 AE 62 36 10 30 60 36 :D0
33D0 10 AD 9F 83 68 16 FF 83 8E 00 00 36 10 8E 00 07 :48
33E0 36 10 8E 00 11 36 10 8E 00 20 36 10 8E 00 23 36 :06
33F0 10 8E 00 07 36 10 8E 00 11 36 10 8E 00 0D 36 10 :B1
-------------------------------------------------------
Sum: 7F 38 C7 CD 27 E6 5B 43 C7 F7 F4 63 64 4F CD C3 :4E


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3400 BD 29 AF BD 34 A0 32 64 39 BD 28 B7 BD 26 8B BD :BC
3410 2A 44 BD 2A 5D BD 2B 7B BD 29 A3 BD 2A 5D BD 2C :CB
3420 25 AE C1 27 06 BD 29 39 16 FF DE 39 37 10 34 10 :97
```

リスト2　Prolog/FV.2マシン語ダンプ・リスト

```
3430 37 10 34 10 BD 32 E0 BD 26 8B BD 28 44 AE C1 27 :87
3440 0C BD 29 39 AE 60 36 10 BD 30 72 20 48 BD 26 8B :B4
3450 8E 80 31 36 10 BD 2A 5D AE C1 27 2F BD 29 39 BD :6A
3460 29 39 8E 00 01 36 10 AE 62 36 10 BD 2A 37 8E 00 :39
3470 00 36 10 8E 00 05 36 10 8E 00 11 36 10 BD 29 AF :99
3480 BD 2F F8 BD 32 E0 BD 29 39 20 0A BD 25 C4 AE 60 :B0
3490 36 10 BD 2F 4B 32 64 39 8E 00 0D 36 10 7E 29 AF :83
34A0 BD 2F F8 7E 29 39 37 10 34 10 37 10 34 10 AE 60 :E8
34B0 36 10 BD 28 59 BD 28 44 AE C1 27 36 AE 62 36 10 :CF
34C0 BD 28 59 BD 28 44 AE C1 27 0F 8E 80 31 36 10 8E :1F
34D0 80 10 36 10 BD 25 C4 20 17 AE 62 36 10 BD 28 59 :47
34E0 BD 26 8B BD 28 54 BD 28 59 BD 29 3C AD 9F 83 73 :49
34F0 20 0F AE 60 36 10 BD 28 59 AE 62 36 10 AD 9F 83 :E6
--------------------------------------------------------
Sum: 06 C2 8B 97 55 79 78 E7 26 B0 10 78 B6 0E 68 73 :14
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3500 73 AE 60 36 10 BD 28 54 BD 2A 44 BD 2A 5D AE C1 :DE
3510 27 02 20 0D AE 60 36 10 BD 28 54 BD 29 3C BD 25 :E7
3520 C4 32 64 39 8E 63 FE 36 10 BD 30 CF BD 26 8B BD :AF
3530 28 54 BD 29 3C BD 34 A6 8E 80 2C 36 10 BD 29 3C :D7
3540 BD 25 C4 8E 63 FC 36 10 BD 2A 37 8E 00 01 36 10 :CC
3550 BD 2A 4A BD 2A 37 39 37 10 34 10 37 10 34 10 AE :4C
3560 62 36 10 BD 28 44 AE C1 27 07 8E 80 10 36 10 20 :F2
3570 2C AE 62 36 10 BD 28 54 BD 26 8B BD 28 54 AE 60 :70
3580 36 10 BD 2A 5D AE C1 27 02 20 12 BD 29 39 AE 60 :81
3590 36 10 AE 62 36 10 BD 28 59 AD 9F 83 8E 32 64 39 :06
35A0 37 10 34 10 37 10 34 10 AE 60 36 10 BD 28 44 BD :50
35B0 2A 50 AE C1 27 7B AE 60 36 10 BD 28 54 BD 26 8B :86
35C0 8E 7F FF 36 10 8E 10 00 36 10 BD 28 BD BD 2C 3F :00
35D0 AE C1 27 2F 8E 7F F0 36 10 AE 62 36 10 BD 28 54 :97
35E0 BD 2A 37 AE 62 36 10 BD 28 54 8E 00 02 36 10 BD :40
35F0 28 BD AE 62 36 10 BD 2A 37 AE 62 36 10 AD 9F 83 :7E
--------------------------------------------------------
Sum: 7C 10 79 B5 74 0D 02 78 AD 17 07 8D 0F E8 A2 D1 :77
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3600 97 20 20 AE 62 36 10 BD 28 54 BD 2A 37 AE 62 36 :CA
3610 10 BD 28 54 8E 00 02 36 10 BD 28 BD AE 62 36 10 :17
3620 BD 2A 37 AE 60 36 10 BD 28 59 37 10 AF 60 16 FF :1B
3630 77 8E 7F F1 36 10 AE 62 36 10 BD 28 54 BD 2A 37 :68
3640 AE 62 36 10 BD 28 54 8E 00 02 36 10 BD 28 BD AE :B5
3650 62 36 10 BD 2A 37 32 64 39 8E 64 00 36 10 37 10 :14
3660 34 10 8E 63 FE 36 10 BD 28 54 BD 26 8B BD 28 44 :49
3670 AE C1 27 0D BD 29 39 8E 83 BA 36 10 BD 25 A3 20 :78
3680 16 30 60 36 10 BD 35 A0 AE 60 36 10 8E 00 01 36 :97
3690 10 BD 28 C4 BD 29 90 32 62 39 37 10 34 10 AE 60 :95
36A0 36 10 BD 28 54 BD 26 8B 8E 00 02 36 10 BD 28 BD :65
36B0 AE 60 36 10 BD 2A 37 BD 28 54 BD 26 8B 8E 7F F1 :17
36C0 36 10 BD 2A 5D AE C1 27 0A BD 29 39 8E 80 10 36 :9D
36D0 10 20 32 BD 26 8B 8E 7F F0 36 10 BD 2A 5D AE C1 :C6
36E0 27 18 BD 29 39 AE 60 36 10 AD 9F 83 A9 AE 60 36 :6E
36F0 10 AD 9F 83 A9 BD 25 C4 20 0B AE 60 36 10 AD 9F :F9
--------------------------------------------------------
Sum: 54 50 BF A3 6B AB 95 09 6A B0 18 BA 17 3D B8 AE :60
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3700 83 A9 BD 25 C4 32 62 39 8E 00 00 36 10 BD 2B 8C :E7
3710 BD 2A 37 8E 64 00 36 10 37 10 34 10 30 60 36 10 :B7
3720 BD 36 9A 8E 63 FE 36 10 BD 2A 37 32 62 39 37 10 :F4
3730 34 10 37 10 34 10 37 10 34 10 37 10 34 10 AE 62 :F5
3740 36 10 AE 64 36 10 AE 66 36 10 AD 9F 83 F3 8E 80 :C8
3750 62 36 10 BD 2A 5D AE C1 27 7C AE 62 36 10 AE 64 :66
3760 36 10 AE 66 36 10 AD 9F 84 20 8E 80 10 36 10 BD :B1
3770 25 C4 BD 26 8B 8E 80 10 36 10 BD 25 C4 AE 60 36 :A5
3780 10 BD 29 3C AD 9F 83 CE AE 66 36 10 BD 28 54 8E :F0
3790 80 84 36 10 BD 2A 5D AE C1 27 35 AE 62 36 10 AE :5D
37A0 64 36 10 AE 66 36 10 AD 9F 83 F3 BD 29 39 BD 26 :C8
37B0 8B AE 62 36 10 AE 64 36 10 AE 66 36 10 AD 9F 84 :63
37C0 2B 8E 80 10 36 10 BD 25 C4 AD 9F 83 CE 16 FF B8 :9F
37D0 BD 29 39 16 FF 68 32 68 39 37 10 34 10 AE 60 36 :3E
37E0 10 BD 28 54 BD 28 44 AE C1 27 14 8E 80 10 36 10 :80
37F0 8E 80 10 36 10 BD 25 C4 AE 60 36 10 BD 2A 37 BD :39
--------------------------------------------------------
Sum: 29 4C B0 DE C2 55 3A 9D 57 2F 05 34 D6 8F 7E 86 :19
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3800 26 8B AE 60 36 10 BD 28 54 BD 31 D0 BD 29 3C 8E :AC
3810 00 00 36 10 BD 29 B5 BD 2F F8 AD 9F 84 3E 8E 00 :61
3820 0A 36 10 BD 29 C1 BD 2F F8 AD 9F 84 3E 8E 00 0D :84
3830 36 10 BD 29 BB BD 2F F8 BD 2B 7B BD 32 44 8E 00 :EF
3840 00 36 10 BD 29 3C 8E 00 20 36 10 BD 29 3C BD 30 :6B
3850 30 8E 00 05 36 10 8E 00 11 36 10 8E 00 01 36 10 :C3
3860 8E 00 0D 36 10 BD 29 AF BD 2F F8 BD 32 E0 BD 25 :0B
3870 C4 BD 2B 7B BD 2A 37 BD 26 8B BD 33 4E 8E 00 00 :7F
3880 36 10 8E 00 20 36 10 8E 00 0D 36 10 8E 00 0A 36 :E9
3890 10 BD 29 B5 BD 2F F8 32 62 39 37 10 34 10 37 10 :2E
38A0 34 10 AE 60 36 10 BD 28 59 BD 28 44 AE C1 27 15 :AA
38B0 AE 62 36 10 AE 60 36 10 8E 00 02 36 10 BD 28 BD :22
38C0 BD 2A 37 20 0F AE 60 36 10 BD 28 59 AE 62 36 10 :35
38D0 AD 9F 83 CE 32 64 39 37 10 34 10 37 10 34 10 37 :B9
38E0 10 34 10 AE 60 36 10 BD 28 54 BD 28 44 AE C1 27 :A0
38F0 41 AE 62 36 10 BD 28 54 BD 28 44 AE C1 27 07 8E :24
--------------------------------------------------------
Sum: CB 3C C0 C0 75 C4 A6 EE 9A 23 9D EB 9D DD A6 14 :CD
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3900 80 67 36 10 20 2A AE 62 36 10 BD 28 54 BD 26 8B :74
3910 BD 28 59 AE 62 36 10 BD 2A 37 BD 28 54 BD 28 59 :29
3920 BD 26 8B BD 28 59 AE 60 36 10 BD 2A 37 BD 28 54 :57
3930 20 17 AE 60 36 10 BD 28 54 BD 26 8B BD 28 59 AE :1E
3940 60 36 10 BD 2A 37 BD 28 54 BD 26 8B BD 2A 44 BD :53
3950 2A 5D BD 2B 7B 8E 81 CF 36 10 BD 2A 5D BD 2C 25 :60
3960 AE C1 27 15 BD 29 39 AE 60 36 10 AE 62 36 10 AE :22
3970 64 36 10 AD 9F 83 F3 20 11 AE 64 36 10 BD 28 54 :2E
3980 BD 29 3C AE 64 36 10 BD 2A 37 32 66 39 37 10 34 :E4
3990 10 37 10 34 10 BD 28 B7 AE 62 36 10 AD 9F 83 C3 :1F
39A0 AE 60 36 10 BD 28 54 BD 28 44 AE C1 27 14 8E 80 :6E
39B0 10 36 10 8E 80 10 36 10 BD 25 C4 AE 60 36 10 BD :71
39C0 2A 37 BD 26 8B BD 28 54 BD 28 54 BD 26 8B AE 60 :BD
39D0 36 10 BD 28 54 BD 28 59 BD 35 57 BD 26 8B BD 28 :59
39E0 44 AE C1 27 26 BD 29 39 8E 80 10 36 10 BD 25 C4 :29
39F0 BD 26 8B 8E 80 10 36 10 BD 25 C4 AE 60 36 10 BD :89
--------------------------------------------------------
Sum: A2 67 24 08 17 AC 04 A3 67 C9 0D E1 51 62 48 07 :BF
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3A00 28 54 BD 29 3C AD 9F 83 CE 20 1C BD 26 8B 8E 80 :F3
3A10 10 36 10 BD 29 3C 8E 00 02 36 10 BD 28 BD BD 2A :D7
3A20 37 BD 29 3C BD 29 39 BD 29 3C BD 28 59 BD 26 8B :46
3A30 BD 28 59 37 10 34 10 BD 28 54 BD 28 59 37 10 34 :BB
3A40 10 30 60 36 10 30 62 36 10 30 7E 34 10 36 10 AD :A3
3A50 9F 83 F3 BD 29 39 30 62 36 10 30 64 36 10 30 60 :76
3A60 36 10 AD 9F 83 D7 32 6A 39 37 10 34 10 37 10 34 :C7
3A70 10 37 10 34 10 AE 64 36 10 BD 28 54 8E 80 67 36 :D7
3A80 10 BD 2A 5D AE 64 36 10 BD 28 54 8E 80 0B 36 10 :44
3A90 BD 2A 5D BD 2C 25 AE 64 36 10 BD 28 54 8E 80 B2 :A3
3AA0 36 10 BD 2A 5D BD 2C 25 AE C1 27 48 AE 60 36 10 :CA
3AB0 AE 62 36 10 AE 64 36 10 AD 9F 83 F3 8E 80 B2 36 :66
3AC0 10 BD 2A 5D AE C1 27 25 AE 60 36 10 AE 62 36 10 :B9
3AD0 AE 64 36 10 AD 9F 84 2B AE 60 36 10 AE 62 36 10 :FD
3AE0 AE 64 36 10 AD 9F 84 20 BD 29 39 20 05 8E 80 10 :AA
3AF0 36 10 20 23 AE 60 36 10 AE 62 36 10 AE 64 36 10 :8B
--------------------------------------------------------
Sum: 74 57 8F 13 99 3D 49 5E C5 FD 22 2B 03 68 F8 28 :84
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3B00 AD 9F 84 2B AE 60 36 10 AE 62 36 10 AE 64 36 10 :FD
3B10 AD 9F 84 20 BD 25 C4 32 66 39 37 10 34 10 37 10 :39
3B20 34 10 37 10 34 10 AE 60 36 10 AE 62 36 10 AE 64 :8B
3B30 36 10 AD 9F 83 F3 BD 26 8B 8E 80 06 36 10 BD 2A :B7
3B40 5D AE C1 27 15 BD 29 39 AE 60 36 10 AE 62 36 10 :D1
3B50 AE 64 36 10 AD 9F 84 20 20 5F BD 26 8B 8E 00 04 :C7
3B60 36 10 BD 28 BD BD 2A 31 8E 00 2A 36 10 BD 2A 5D :42
3B70 AE C1 27 02 20 43 AE 64 36 10 BD 28 54 8E 80 62 :FC
3B80 36 10 BD 2A 5D AE C1 27 28 AE 60 36 10 AE 62 36 :E2
3B90 10 AE 64 36 10 AD 9F 83 F3 BD 29 39 AE 60 36 10 :9D
3BA0 AE 62 36 10 AE 64 36 10 AD 9F 84 20 BD 2C D1 20 :78
3BB0 08 8E 80 10 36 10 BD 2C D1 32 66 39 8E 00 00 36 :BB
3BC0 10 8E 00 20 36 10 8E 00 28 36 10 8E 00 29 36 10 :FD
3BD0 8E 00 2C 36 10 8E 00 2D 36 10 8E 00 2E 36 10 8E :91
3BE0 00 5B 36 10 8E 00 5D 36 10 39 30 8C 08 36 10 BE :D3
3BF0 84 68 6E 88 11 90 00 30 8C 08 36 10 BE 84 68 6E :A5
--------------------------------------------------------
Sum: D1 40 6E C9 F7 E1 28 2F FA CB EC 0E E8 22 DF E7 :06
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3C00 88 11 90 04 30 8C 08 36 10 BE 84 68 6E 88 11 90 :78
3C10 08 30 8C 08 36 10 BE 84 68 6E 88 11 90 0C 30 8C :1B
3C20 08 36 10 BE 84 68 6E 88 11 2B BD BD 2C 50 8E 00 :AE
3C30 02 36 10 BD 2B DE 8E 84 68 BD 2C 87 39 37 10 34 :AC
3C40 10 37 10 34 10 37 10 34 10 37 10 34 10 AE 62 36 :F7
3C50 10 AE 64 36 10 AE 66 36 10 BD 38 D7 8E 80 84 36 :56
3C60 10 BD 2A 5D AE C1 27 21 AE 60 36 10 AE 62 36 10 :B5
3C70 AE 64 36 10 AE 66 36 10 BD 3B 1A 8E 80 10 36 10 :28
3C80 BD 25 C4 BD 38 9A 16 FF C4 32 68 39 37 10 34 10 :6C
3C90 37 10 34 10 AE 60 36 10 AE 62 36 10 BD 39 8D AE :66
3CA0 60 36 10 BD 28 54 BD 3B EA BD 2A 37 AE 62 36 10 :35
3CB0 BD 28 54 BD 3C 04 BD 2A 37 32 64 39 37 10 34 10 :AE
3CC0 37 10 34 10 AE 60 36 10 AE 62 36 10 BD 39 8D AE :66
3CD0 60 36 10 BD 28 54 BD 3B F7 BD 2A 37 AE 62 36 10 :42
3CE0 BD 28 54 BD 3C 11 BD 2A 37 32 64 39 37 10 34 10 :BB
3CF0 37 10 34 10 7E 40 39 AE 60 8C 80 10 27 06 12 12 :FD
--------------------------------------------------------
Sum: 14 C4 38 3F 6B 45 44 F8 4B 03 FD AF D1 27 65 9A :2C
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3D00 AE C1 27 6A AE C1 27 48 AE 60 36 10 BD 28 54 AE :19
3D10 62 36 10 AD 9F 84 D0 AE 60 36 10 BD 28 59 BD 28 :BF
3D20 44 AE C1 27 02 20 27 AE 60 36 10 BD 28 59 37 10 :FC
3D30 AF 60 8E 80 62 36 10 BD 25 A3 AE 60 36 10 AE 62 :AE
3D40 36 10 AD 9F 84 F7 8E 80 67 36 10 BD 25 A3 20 1C :89
3D50 8E 80 06 36 10 BD 25 A3 AE 60 36 10 AE 62 36 10 :89
3D60 AD 9F 84 F7 8E 80 0B 36 10 BD 25 A3 20 07 AE 60 :E0
3D70 36 10 BD 25 A3 32 64 39 37 10 34 10 37 10 34 10 :B0
3D80 30 60 36 10 30 62 36 10 AD 9F 84 ED AE C1 27 0E :0F
3D90 AE C1 27 02 20 08 8E 81 CF 36 10 BD 25 A3 AE 60 :77
3DA0 36 10 AE 62 36 10 AD 9F 84 F7 32 64 39 37 10 34 :AD
3DB0 10 37 10 34 10 AE 60 36 10 AE 62 36 10 AD 9F 84 :15
3DC0 DA AE 60 36 10 BD 28 54 8E 7F FF 36 10 8E 10 00 :57
3DD0 36 10 BD 28 BD BD 2C 3F BD 26 8B AE C1 27 0D AE :CF
3DE0 60 36 10 BD 28 54 BD 2C D7 BD 29 3C 32 64 39 37 :C7
```

リスト2　Prolog/FV.2マシン語ダンプ・リスト

```
3DF0 10 34 10 37 10 34 10 30 60 36 10 30 62 36 10 AD :3A
----------------------------------------------------------
Sum: 4E D4 D2 A9 11 2B 42 48 81 E4 8E FE EE 9D 18 9C :93


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3E00 9F 84 ED AE C1 27 3A AE C1 27 16 8E 80 B2 36 10 :92
3E10 BD 25 A3 AE 60 36 10 AE 62 36 10 AD 9F 84 D0 20 :EF
3E20 1E AE 60 36 10 BD 28 54 AE 62 36 10 AD 9F 84 D0 :A1
3E30 AE 60 36 10 BD 28 59 AE 62 36 10 AD 9F 84 E3 20 :BB
3E40 21 AE 60 36 10 BD 28 44 AE C1 27 02 20 14 8E 80 :78
3E50 B2 36 10 BD 25 A3 AE 60 36 10 AE 62 36 10 AD 9F :73
3E60 84 D0 32 64 39 37 10 34 10 37 10 34 10 AE 60 36 :7D
3E70 10 BD 28 54 AE 62 36 10 BD 28 54 BD 28 59 BD 35 :08
3E80 57 BD 26 8B BD 28 44 AE C1 27 05 BD 29 39 20 26 :EE
3E90 BD 28 59 BD 26 8B BD 28 54 AE 60 36 10 BD 2A 37 :57
3EA0 BD 28 59 AE 62 36 10 BD 2A 37 AE 60 36 10 AE 62 :16
3EB0 36 10 AD 9F 84 DA 32 64 39 37 10 6E 94 37 10 34 :83
3EC0 10 AE 60 36 10 BD 22 C0 BD 26 8B BD 28 44 BD 2A :81
3ED0 50 AE C1 27 12 BD 26 8B BD 28 54 BD 28 54 BD 28 :BD
3EE0 54 BD 25 A3 16 FF DE BD 29 39 32 62 39 30 8C 08 :7C
3EF0 36 10 BE 84 68 6E 88 11 90 10 30 8C 08 36 10 BE :5F
----------------------------------------------------------
Sum: 80 6E 79 66 73 E5 D8 56 8F FF 09 76 8D BF E3 B5 :44


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3F00 84 68 6E 88 11 00 01 8E 80 10 36 10 8E 80 10 36 :AC
3F10 10 BD 25 C4 BD 26 8B 12 12 12 12 BD 3B EA BD 2A :35
3F20 37 8E 80 10 36 10 8E 80 10 36 10 BD 25 C4 BD 26 :88
3F30 8B 12 12 12 12 BD 3B F7 BD 2A 37 8E 1F FF 36 10 :D2
3F40 BD 2A 31 AE C1 27 1D 8E 64 00 36 10 37 10 34 10 :8E
3F50 30 60 36 10 BD 36 9A 30 60 36 10 BD 36 9A AD 9F :12
3F60 85 EB 20 1A 8E 80 10 36 10 36 10 12 32 7E BD 25 :F8
3F70 C4 8E 80 10 36 10 8E 80 10 36 10 BD 25 C4 BD 2B :1A
3F80 7B BD 3C 04 BD 2A 37 BD 26 8B BD 3C 11 BD 2A 37 :2C
3F90 8E 80 10 36 10 8E 80 10 36 10 BD 25 C4 BD 26 8B :DC
3FA0 BD 3E ED BD 2A 37 8E 00 01 36 10 BD 3E FA BD 2A :B7
3FB0 37 12 12 12 12 32 62 39 BD 28 B7 BD 26 8B 8E 83 :67
3FC0 FE 36 10 BD 2A 5D AE C1 27 1A BD 29 39 BD 28 B7 :F3
3FD0 BD 3C 04 BD 28 54 BD 37 D9 8E 83 FE 36 10 AD 9F :A4
3FE0 85 B8 20 4B BD 26 8B 8E 84 0B 36 10 BD 2A 5D AE :6B
3FF0 C1 27 1A BD 29 39 BD 28 B7 BD 3C 11 BD 28 54 BD :BD
----------------------------------------------------------
Sum: 8A A6 C5 E1 99 11 04 3F 98 8D E8 D7 F3 37 3C C5 :D2


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4000 37 D9 8E 84 0B 36 10 AD 9F 85 B8 20 22 BD 26 8B :AC
4010 8E 84 36 36 10 BD 2A 5D AE C1 27 0F BD 29 39 BD :53
4020 3E ED BD 28 54 AD 9F 84 36 20 04 AD 9F 86 18 8E :06
4030 00 00 36 10 AE C1 27 80 39 30 60 36 10 30 62 36 :33
4040 10 AD 9F 84 ED 7E 3C F7 36 10 BD 2A 5D AE C1 27 :9E
4050 19 BD 29 39 8E 00 01 36 10 BD 3E FA BD 28 54 BD :F8
4060 28 C4 BD 3E FA BD 2A 37 20 3F 8E 82 83 36 10 BD :F4
4070 2A 5D AE C1 27 2B 30 60 36 10 BD 3B EA BD 28 54 :39
4080 BD 35 A0 30 60 36 10 BD 3B F7 BD 28 54 BD 35 A0 :22
4090 AE 60 36 10 8E 00 01 36 10 BD 28 C4 BD 29 90 20 :68
40A0 08 8E 84 8A 36 10 BD 25 A3 32 62 39 37 10 34 10 :C7
40B0 37 10 34 10 AE 60 36 10 BD 28 54 BD 28 54 BD 26 :34
40C0 8B AE 62 36 10 BD 3E B9 BD 28 54 BD 28 59 BD 35 :FE
40D0 57 BD 26 8B BD 28 44 AE C1 27 28 BD 29 39 8E 80 :D9
40E0 10 36 10 BD 25 C4 BD 26 8B 8E 80 10 36 10 BD 25 :B0
40F0 C4 AE 62 36 10 BD 3E B9 BD 28 54 BD 29 3C BD 38 :1E
----------------------------------------------------------
Sum: DE 57 72 3C 8D D3 18 40 C9 C5 74 1C 35 8D A1 09 :25


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4100 9A 20 1C BD 29 3C BD 29 39 BD 26 8B 8E 80 10 36 :D9
4110 10 BD 29 3C 8E 00 02 36 10 BD 28 BD BD 2A 37 AE :76
4120 60 36 10 BD 28 59 BD 26 8B BD 28 59 37 10 34 10 :1B
4130 BD 28 54 BD 28 59 37 10 34 10 30 60 36 10 30 62 :6A
4140 36 10 30 7E 34 10 36 10 BD 38 D7 BD 29 39 30 62 :FB
4150 36 10 30 64 36 10 30 60 36 10 BD 37 2E 32 6A 39 :ED
4160 AE C4 34 10 AE 42 34 10 AE F4 36 10 BD 22 C0 BD :2E
4170 26 8B BD 28 44 BD 2A 50 AE C1 27 12 BD 26 8B BD :E4
4180 28 54 8E 83 FE 36 10 AD 9F 85 B8 16 FF DE BD 29 :33
4190 39 AE D4 36 10 BD 22 C0 BD 26 8B BD 28 44 BD 2A :1E
41A0 50 AE C1 27 12 BD 26 8B BD 28 54 8E 84 0B 36 10 :02
41B0 AD 9F 85 B8 16 FF DE BD 29 39 32 64 39 AE D4 EC :D8
41C0 D8 02 AE 02 8C 80 10 27 07 10 A3 94 27 05 20 F2 :59
41D0 33 44 39 AE 84 EC 98 02 AE 02 AE 02 AF D4 ED D8 :10
41E0 02 20 DA 37 10 34 10 8E 80 10 36 10 AE 60 36 10 :3F
41F0 BD 37 D9 BD 28 59 BD 26 8B BD 28 59 37 10 34 10 :42
----------------------------------------------------------
Sum: 2F 96 3C C9 E1 B5 22 F7 59 2F 0F DB 28 A1 8B A4 :E3


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4200 BD 28 54 BD 28 59 37 10 34 10 30 60 36 10 30 62 :6A
4210 36 10 30 7E 34 10 36 10 BD 38 D7 BD 29 39 8E 80 :77
4220 10 36 10 8E 80 10 36 10 BD 25 C4 BD 26 8B 30 62 :60
4230 36 10 30 64 36 10 30 60 36 10 BD 3C 3D BD 28 59 :6A
4240 8E 80 10 36 10 8E 80 10 36 10 BD 25 C4 BD 2B 7B :D1
4250 BD 2B 7B 8E 80 10 36 10 AD 9F 85 2B 8E 83 2B 36 :35
4260 10 BD 2A 5D AE C1 27 0A 8E 86 23 36 10 BD 25 A3 :F6
4270 20 08 8E 86 34 36 10 BD 25 A3 BD 29 3C BD 26 8B :CB
4280 BD 28 44 AE C1 27 08 BD 29 39 BD 29 39 20 10 BD :F2
4290 28 54 BD 29 3C AD 9F 84 C7 BD 2A 44 BD 25 A3 32 :17
42A0 68 39 37 10 34 10 37 10 34 10 8E 00 01 36 10 BD :49
42B0 29 A9 8E 00 02 36 10 BD 28 BD BD 2A 37 AE 60 36 :AC
42C0 10 AE 62 36 10 BD 3C EC 8E 00 00 36 10 BD 29 A9 :AE
42D0 8E 00 02 36 10 BD 28 BD BD 2A 37 32 64 39 37 10 :AC
42E0 34 10 37 10 34 10 37 10 34 10 37 10 34 10 37 10 :2C
42F0 34 10 30 60 36 10 30 62 36 10 BD 3D AD AE C1 27 :2F
----------------------------------------------------------
Sum: 30 1A 98 97 41 D2 79 A0 7B 62 07 11 E3 28 32 4E :25


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4300 46 30 64 36 10 30 66 36 10 BD 3D AD AE C1 27 1A :53
4310 AE 60 36 10 AE 62 36 10 AE 64 36 10 AE 66 36 10 :5C
4320 AE 68 36 10 AD 9F 85 12 20 1B BD 29 39 AE 64 36 :E1
4330 10 AE 66 36 10 AE 60 36 10 AE 62 36 10 AE 68 36 :60
4340 10 AD 9F 85 0A 20 23 30 64 36 10 30 66 36 10 BD :A1
4350 41 BD AE 60 36 10 AE 62 36 10 AE 64 36 10 AE 66 :14
4360 36 10 AE 68 36 10 AD 9F 85 0A 32 6A 39 37 10 34 :CD
4370 10 37 10 34 10 37 10 34 10 37 10 34 10 37 10 34 :2C
4380 10 BD 2B 7B BD 2A 5D AE C1 27 4D BD 29 39 AE 60 :C7
4390 36 10 BD 28 54 AE 62 36 10 AE 64 36 10 BD 28 54 :66
43A0 AE 66 36 10 AE 68 36 10 BD 42 DE BD 28 44 AE C1 :2B
43B0 27 07 8E 80 10 36 10 20 1D AE 60 36 10 BD 28 59 :61
43C0 AE 62 36 10 AE 64 36 10 BD 28 59 AE 66 36 10 AE :F4
43D0 68 36 10 BD 42 DE 20 2E AE C1 27 12 AE 60 36 10 :D5
43E0 AE 62 36 10 AE 64 36 10 AE 66 36 10 20 10 AE 64 :4A
43F0 36 10 AE 66 36 10 AE 60 36 10 AE 62 36 10 AE 68 :60
----------------------------------------------------------
Sum: 5E 9B 17 83 A4 82 4E B5 17 95 E5 66 65 E4 55 79 :CA


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4400 36 10 AD 9F 86 45 32 6A 39 37 10 34 10 37 10 34 :38
4410 10 37 10 34 10 37 10 34 10 37 10 34 10 AE 60 36 :F5
4420 10 AD 9F 85 8B AE C1 27 6B AE 60 36 10 AE 64 36 :09
4430 10 BD 2A 5D AE 62 36 10 AE 66 36 10 BD 2A 5D BD :05
4440 2C 13 AE C1 27 07 8E 83 2B 36 10 20 45 AE 60 36 :07
4450 10 AE 64 36 10 AE 66 36 10 BD 25 C4 BD 25 C4 AE :BC
4460 62 36 10 BD 28 59 BD 25 C4 AE 62 36 10 8E 00 02 :72
4470 36 10 BD 28 BD BD 2A 37 AE 62 36 10 AE 68 36 10 :B8
4480 BD 28 54 BD 25 C4 AE 68 36 10 BD 2A 37 8E 83 2B :95
4490 36 10 20 41 AE 60 36 10 AE 64 36 10 BD 2A 5D AE :45
44A0 C1 27 07 8E 83 2B 36 10 20 2B AE 64 36 10 AD 9F :60
44B0 85 8B AE C1 27 1A AE 64 36 10 AE 66 36 10 AE 60 :80
44C0 36 10 AE 62 36 10 AE 68 36 10 AD 9F 85 0A 20 05 :F8
44D0 8E 80 10 36 10 32 6A 39 EF C3 39 EE C4 39 37 10 :56
44E0 34 10 AE 60 36 10 8E 7F FF 36 10 BD 2C 3F AE C1 :81
44F0 27 19 AE 60 36 10 8E 00 04 36 10 BD 28 BD BD 2A :F5
----------------------------------------------------------
Sum: 92 5B A8 36 1A 22 10 F6 71 73 D8 E3 AA 9D 88 2B :A6


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4500 31 8E 00 2A 36 10 BD 2A 5D 20 05 8E 00 00 36 10 :6C
4510 32 62 39 37 10 34 10 8E 7F FF 36 10 BD 26 8B AE :C6
4520 60 36 10 BD 2C 2E BD 29 3C 8E 10 00 36 10 BD 28 :A8
4530 BD AE 60 36 10 BD 2C 3F BD 2C 13 32 62 39 37 10 :49
4540 34 10 37 10 34 10 37 10 34 10 37 10 34 10 BD 3E :E0
4550 FA BD 28 54 8E 00 00 36 10 BD 2A 5D AE C1 27 7D :5E
4560 AE 60 36 10 AE 62 36 10 BD 42 A2 AE 64 36 10 BD :60
4570 28 59 BD 26 8B BD 28 44 AE C1 27 0D BD 29 39 8E :68
4580 86 55 36 10 BD 25 A3 20 54 8E 86 63 36 10 BD 25 :B9
4590 A3 BD 26 8B BD 29 A9 8E 00 02 36 10 BD 28 BD BD :D5
45A0 2A 37 8E 80 84 36 10 BD 25 A3 BD 26 8B BD 28 54 :65
45B0 AE 66 36 10 BD 42 A2 BD 28 59 BD 26 8B BD 28 44 :D0
45C0 BD 2A 50 AE C1 27 0B 8E 86 6D 36 10 BD 25 A3 16 :3A
45D0 FF BF BD 29 39 8E 27 D2 BD 21 67 12 12 AE 64 36 :15
45E0 10 BD 28 59 AE 66 36 10 32 68 39 37 10 34 10 AE :B4
45F0 60 36 10 BD 28 54 BD 28 44 BD 2A 50 AE C1 27 35 :0A
----------------------------------------------------------
Sum: B1 E5 60 06 08 93 6E 7A DE E8 BE 60 EE 19 EA A5 :F9


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4600 AE 60 36 10 BD 28 54 BD 28 54 BD 26 8B BD 28 59 :72
4610 BD 28 59 BD 29 3C 8E 00 02 36 10 BD 28 BD BD 2A :BF
4620 37 AE 60 36 10 BD 28 54 BD 28 59 AE 60 36 10 BD :13
4630 2A 37 16 FF BA 32 62 39 37 10 34 10 37 10 34 10 :13
4640 37 10 34 10 8E 80 10 36 10 37 10 34 10 8E 80 10 :98
4650 36 10 8E 80 10 36 10 BD 25 C4 37 10 34 10 AE 64 :ED
4660 36 10 BD 28 44 AE C1 27 33 AE 68 36 10 BD 28 44 :BD
4670 AE C1 27 07 8E 83 2B 36 10 20 1F AE 68 36 10 BD :77
4680 28 54 BD 28 54 AE 68 36 10 BD 28 54 BD 28 59 AE :36
4690 68 36 10 BD 28 59 AD 9F 85 2B 20 75 AE 64 36 10 :D5
46A0 AE 66 36 10 AD 9F 86 C2 BD 28 54 BD 29 3C BD 29 :2F
46B0 39 BD 3B EA BD 28 54 BD 28 59 BD 35 57 BD 26 8B :49
46C0 BD 28 44 AE C1 27 19 BD 29 39 AE 64 36 10 AE 66 :63
46D0 36 10 AE 68 36 10 30 62 36 10 AD 9F 85 77 20 1B :FD
46E0 BD 28 59 AE E4 36 10 AE 64 36 10 AE 66 36 10 AE :76
46F0 68 36 10 30 62 36 10 6E 9F 85 3C BD 26 8B AE E4 :54
----------------------------------------------------------
Sum: AC A1 44 94 43 AB D0 29 72 F8 28 F2 38 1E 8D 4A :BD


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4700 36 10 BD 2A 5D AE C1 27 08 BD 29 39 8E 80 10 36 :9B
4710 10 32 6A 39 37 10 34 10 37 10 34 10 37 10 34 10 :86
4720 37 10 34 10 37 10 34 10 37 10 34 10 AE 60 36 10 :F5
```

リスト2　Prolog/FV.2 マシン語ダンプ・リスト

```
4730 BD 28 44 AE C1 27 0A 8E 80 10 36 10 32 6C 7E 46 :8F
4740 FB 7E 47 75 AE 64 AE 02 8C 80 10 27 12 36 10 AE :40
4750 66 36 10 BD 25 C4 AE 68 36 10 BD 25 C4 20 04 AE :26
4760 68 36 10 AE 60 AE 02 8C 80 10 27 03 7E 47 F9 32 :A2
4770 6C 7E 47 B8 10 AE 64 36 10 AE 66 36 10 AD 9F 86 :7D
4780 C2 BD 28 59 BD 29 3C AE 60 36 10 BD 28 54 BD 28 :94
4790 54 AE 62 36 10 AE 6A 36 10 BD 42 DE BD 28 44 AE :BC
47A0 C1 27 28 AE 6A 36 10 BD 45 EB AE 60 36 10 BD 28 :94
47B0 59 37 10 AF 60 7E 47 2C BD 46 38 AE C4 8C 83 2B :87
47C0 27 07 30 62 36 10 BD 45 EB 20 65 AE 64 36 10 BD :8D
47D0 28 54 AE 66 36 10 AE 60 36 10 BD 28 54 AE 62 36 :A9
47E0 10 BD 45 3E 7E 47 44 02 8C 80 10 27 00 36 10 BD :A1
47F0 25 C4 AE 68 36 10 BD 25 C4 AD 9F 85 2B BD 26 8B :55
-------------------------------------------------------
Sum: 23 87 E0 13 86 7B 5E 9A 2B BC 2A 19 CB 95 8D 14 :C1
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4800 8E 83 2B 36 10 BD 2A 5D AE C1 27 02 20 33 AE 6A :C9
4810 36 10 BD 45 EB BD 26 8B BD 28 44 AE C1 27 22 BD :3F
4820 29 39 AE 60 36 10 BD 28 59 37 10 AF 60 7E 47 2C :3B
4830 7E 46 FB 36 10 35 10 36 10 35 10 36 10 6E 9F 85 :AD
4840 3C 7E 47 3C 37 10 34 10 37 10 34 10 37 10 34 10 :DE
4850 37 10 34 10 AE 60 36 10 BD 28 54 BD 28 54 BD 26 :34
4860 8B 8E 85 93 36 10 BD 2A 5D AE C1 27 53 BD 29 39 :C3
4870 AE 60 36 10 BD 28 54 BD 28 59 BD 28 54 AE 62 36 :4A
4880 10 AE 66 36 10 AD 9F 85 99 AE C1 27 25 AE 60 36 :D3
4890 10 BD 28 59 AE 62 36 10 AE 64 36 10 AD 9F 85 2B :F8
48A0 AE 60 36 10 AE 62 36 10 AE 66 36 10 AD 9F 86 75 :4B
48B0 20 0C AE 66 36 10 BD 45 EB 8E 80 10 36 10 20 59 :50
48C0 8E 81 05 36 10 BD 2A 5D AE C1 27 26 AE 60 36 10 :AE
48D0 BD 28 59 AE 62 36 10 AE 64 36 10 AD 9F 85 2B BD :A5
48E0 26 8B BD 28 44 AE C1 27 07 BD 29 39 AE 62 36 10 :EC
48F0 20 27 AE 60 36 10 BD 28 54 AE 62 36 10 BD 42 A2 :CB
-------------------------------------------------------
Sum: 96 C0 02 71 A7 99 18 91 9A FC 00 4A 17 15 96 2B :7F
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4900 BD 2A 44 BD 25 A3 AE 60 36 10 BD 28 59 AE 62 36 :88
4910 10 AE 64 36 10 AD 9F 85 2B 32 68 39 37 10 34 10 :C2
4920 37 10 34 10 37 10 34 10 BD 44 D8 37 10 34 10 8E :08
4930 00 01 36 10 37 10 34 10 30 64 36 10 30 66 36 10 :88
4940 BD 3D AD AE 60 36 10 BD 2C 13 AE C1 27 25 BD 29 :98
4950 39 AE 64 36 10 AE 66 36 10 AE 68 36 10 30 60 36 :0D
4960 10 AD 9F 86 B7 AE 64 36 10 BD 28 59 37 10 AF 64 :89
4970 16 FF C5 AE 60 36 10 8E 00 00 36 10 BD 2A 5D AE :F4
4980 C1 27 11 AE 62 36 10 BD 44 DB BD 29 39 8E 00 00 :D8
4990 36 10 20 28 BD 44 D8 AE 62 36 10 8E 00 02 36 10 :93
49A0 BD 28 C4 BD 2A 5D BD 2A 50 AE C1 27 0F AE 62 36 :0F
49B0 10 BD 44 DB BD 29 39 8E 00 01 36 10 32 6A 39 37 :EC
49C0 10 34 10 37 10 34 10 37 10 34 10 BD 26 8B 8E 83 :E9
49D0 2B 36 10 BD 2A 5D AE C1 27 02 20 23 AE 60 36 10 :E4
49E0 BD 28 54 BD 28 59 BD 28 59 AE 62 36 10 AE 64 36 :53
49F0 10 AD 9F 85 99 BD 29 39 AE 64 36 10 BD 45 EB 32 :10
-------------------------------------------------------
Sum: EC DB D3 CF 2B DF 21 38 CE 70 33 1C 16 6D E9 CD :92
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4A00 66 39 37 10 34 10 37 10 34 10 37 10 34 10 37 10 :87
4A10 34 10 AE 60 36 10 BD 28 54 37 10 34 10 AE 68 36 :A8
4A20 10 BD 28 54 8E 00 01 36 10 BD 2A 5D AE C1 27 67 :5F
4A30 30 60 36 10 30 64 36 10 BD 3D AD AE C1 27 4A BD :F4
4A40 29 39 AE 60 36 10 BD 28 54 BD 26 8B 37 10 AF 60 :B3
4A50 BD 45 13 AE C1 27 2C AE 60 36 10 8E 80 26 36 10 :A5
4A60 BD 2A 5D AE C1 27 0E 8E 00 02 36 10 AE 68 36 10 :1A
4A70 BD 2A 37 20 0C AE 60 36 10 AE 68 36 10 AD 9F 86 :CC
4A80 CC 20 04 AE 60 36 10 20 0C 8E 00 00 36 10 AE 68 :5A
4A90 36 10 BD 2A 37 20 61 AE 68 36 10 BD 28 54 8E 00 :08
4AA0 02 36 10 BD 2A 5D AE C1 27 3E 8E 80 10 36 10 BD :81
4AB0 2C D1 AE 64 36 10 AE 60 36 10 AE 64 36 10 AE 66 :15
4AC0 36 10 BD 42 DE BD 28 44 AE C1 27 0E 8E 00 00 36 :B4
4AD0 10 AE 68 36 10 BD 2A 37 20 0C 8E 00 01 36 10 AE :39
4AE0 68 36 10 BD 2A 37 20 10 AE 60 36 10 8E 00 01 36 :15
4AF0 10 AE 68 36 10 BD 2A 37 32 6A 39 37 10 34 10 37 :21
-------------------------------------------------------
Sum: 28 11 B4 14 0B C1 EB C9 98 8D 62 A4 F9 05 E5 4C :DB
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4B00 10 34 10 AE 60 36 10 8E 81 DA 36 10 BD 2A 5D AE :C9
4B10 C1 27 0E 8E 00 03 36 10 AE 62 36 10 BD 2A 37 20 :61
4B20 07 AE 60 36 10 BD 3E B9 32 64 39 37 10 34 10 37 :A0
4B30 10 34 10 AE 60 36 10 BD 28 54 37 10 AF 60 30 60 :C7
4B40 36 10 30 62 36 10 BD 41 BD AE 62 36 10 AE 60 36 :73
4B50 10 32 64 39 37 10 34 10 37 10 34 10 37 10 34 10 :80
4B60 37 10 34 10 37 10 34 10 AE 60 36 10 AE 62 36 10 :C0
4B70 BD 4B 2B BD 29 3C BD 29 39 BD 3B F7 BD 28 54 BD :59
4B80 28 59 BD 35 57 37 10 34 10 8E 80 10 36 10 37 10 :00
4B90 34 10 AE 62 36 10 BD 28 44 AE C1 27 07 8E 80 10 :7E
4BA0 36 10 20 70 AE 62 36 10 BD 28 59 8E 80 10 36 10 :CE
4BB0 8E 80 10 36 10 BD 25 C4 AE 64 36 10 AE 66 36 10 :BC
4BC0 AE 68 36 10 AE 6A 36 10 30 60 36 10 AD 9F 86 D7 :39
4BD0 BD 26 8B 8E 83 2B 36 10 BD 2A 5D AE C1 27 1B AE :93
4BE0 60 36 10 AE 6C 36 10 BD 28 54 BD 3B 9A AE 60 36 :12
4BF0 10 AE 6C 36 10 BD 2A 37 20 1A BD 28 44 AE C1 27 :87
-------------------------------------------------------
Sum: 1D 45 59 47 95 86 44 E2 58 8F C0 A7 A2 66 D7 9A :0A
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4C00 07 8E 80 10 36 10 20 0C 30 60 36 10 BD 45 EB 8E :E8
4C10 80 10 36 10 32 6E 39 37 10 34 10 37 10 34 10 37 :FC
4C20 10 34 10 37 10 34 10 37 10 34 10 37 10 34 10 37 :2C
4C30 10 34 10 AE 60 36 10 BD 28 54 BD 28 54 AE 62 36 :60
4C40 10 AE 64 36 10 BD 28 59 AE 66 36 10 AE 6C 36 10 :60
4C50 BD 42 DE BD 26 8B BD 28 44 AE C1 27 02 20 47 BD :30
4C60 29 39 AE 60 36 10 BD 28 54 BD 28 59 BD 28 59 AE :19
4C70 62 36 10 8E 80 10 36 10 BD 46 38 AD 9F 86 E6 AE :AD
4C80 C1 27 23 BD 29 39 AE 60 36 10 BD 28 54 BD 28 59 :F5
4C90 BD 28 54 AE 62 36 10 AE 68 36 10 AE 6A 36 10 AE :F7
4CA0 6C 36 10 BD 42 DE BD 26 8B BD 28 44 AE C1 27 42 :FE
4CB0 BD 29 39 AE 6C 36 10 BD 45 EB AE 60 36 10 BD 28 :A5
4CC0 59 BD 28 44 AE C1 27 07 8E 80 10 36 10 20 23 35 :FB
4CD0 10 36 10 BD 28 59 35 10 36 10 35 10 36 10 35 10 :EF
4CE0 36 10 35 10 36 10 35 10 36 10 35 10 36 10 6E 9F :F4
4CF0 86 D7 32 6E 39 37 10 34 10 AE 60 36 10 AE 60 36 :59
-------------------------------------------------------
Sum: CB ED 35 3B 42 34 7D 3C F3 6F E7 E9 6B 47 6B E6 :8C
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4D00 10 8E 83 2B 36 10 BD 2A 5D 32 62 39 8E 64 00 36 :CB
4D10 10 37 10 34 10 BD 26 8B 8E 85 AA 36 10 BD 2A 5D :50
4D20 AE C1 27 19 BD 29 39 8E 00 01 36 10 BD 3E FA BD :55
4D30 28 54 BD 28 C4 BD 3E FA BD 2A 37 20 75 BD 26 8B :3B
4D40 8E 82 83 36 10 BD 2A 5D AE C1 20 31 BD 29 39 BD :B9
4D50 3C 04 BD 28 54 30 60 36 10 BD 35 A0 BD 3C 11 BD :A8
4D60 28 54 30 60 36 10 BD 35 A0 AE 60 36 10 8E 00 01 :C7
4D70 36 10 BD 28 C4 BD 29 90 BD 2D EF 20 35 8E 84 53 :F8
4D80 AC C4 26 26 33 42 12 12 12 BD 3E ED BD 28 54 BD :45
4D90 28 8B 8E 00 00 36 10 BD 28 BD 8E 80 10 36 10 BD :48
4DA0 29 3C BD 2A 37 BD 41 E3 20 0B BD 27 8A 12 12 12 :30
4DB0 12 12 32 62 39 BD 29 0A BD 29 39 39 8E 1C FF 36 :18
4DC0 10 BD 2A 3D 30 8C 10 34 10 34 76 10 FF 1F 14 B6 :E6
4DD0 FD 0F 10 FE 1F 12 39 F6 1C FE 4F 36 06 39 25 A3 :20
4DE0 01 02 70 72 25 AF 11 01 5B 25 BD 23 01 5D 25 16 :C4
4DF0 05 03 6E 69 6C 25 C4 01 04 63 6F 6E 73 26 8B 01 :9E
-------------------------------------------------------
Sum: 3E 32 5F 4E A8 D1 74 7D 65 A0 D0 6A ED 04 76 DB :08
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4E00 03 64 75 70 26 90 19 02 3D 3A 26 F5 09 01 3A 27 :1A
4E10 1D 10 01 3B 25 16 10 02 69 66 25 16 10 04 65 6C :A5
4E20 73 65 25 16 10 04 6C 69 73 74 28 44 01 04 6E 75 :37
4E30 6C 6C 28 54 01 03 63 61 72 28 59 01 03 63 64 72 :4C
4E40 28 60 11 01 28 28 73 03 01 29 28 A7 01 04 65 63 :26
4E50 68 6F 28 B7 01 07 67 65 74 77 6F 72 64 28 BD 01 :A0
4E60 01 2B 28 C4 01 2D 28 CB 01 01 2A 29 01 01 01 :92
4E70 2F 29 0A 01 04 2F 6D 6F 64 29 39 01 04 64 72 6F :82
4E80 70 29 3C 01 04 73 77 61 70 29 43 01 03 72 6F 74 :5A
4E90 29 4A 01 01 2E 29 90 01 04 2E 68 65 78 29 A3 01 :A1
4EA0 02 63 72 29 A9 01 04 62 61 73 65 29 AF 01 06 70 :98
4EB0 72 6F 6D 70 74 29 B5 01 06 69 67 6E 6F 72 63 29 :C2
4EC0 BB 01 06 64 65 6C 69 6D 63 29 C1 01 05 74 61 69 :5E
4ED0 6C 63 29 C7 19 02 2A 40 28 54 01 01 40 2A 31 01 :5E
4EE0 02 63 40 2A 37 01 01 21 2A 3D 01 02 63 21 2A 44 :85
4EF0 01 02 73 70 2A 4A 01 06 73 77 69 74 63 68 25 16 :2E
-------------------------------------------------------
Sum: F6 76 2C F2 B8 8B C1 66 32 6A 40 09 53 32 62 20 :E0
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4F00 01 08 65 64 69 74 6C 69 73 74 25 16 01 05 63 6C :7B
4F10 69 73 74 25 16 01 02 0D 0A 25 16 01 01 20 2A 50 :7C
4F20 01 03 6E 6F 74 2A 5D 01 01 3D 2A 6D 10 06 72 65 :9F
4F30 70 65 61 74 2A 89 10 05 75 6E 74 69 6C 2A DA 01 :A3
4F40 07 70 72 69 6E 74 65 72 2A E0 01 06 6E 75 6D 62 :CE
4F50 65 72 2A E6 10 05 77 68 69 6C 65 2B 2E 10 03 65 :E6
4F60 6E 64 2B 7B 01 04 6F 76 65 72 2B 80 01 06 65 78 :C8
4F70 70 61 6E 64 2B 86 01 07 73 61 76 65 65 6E 64 2B :6D
4F80 8C 01 09 64 75 6D 70 70 6F 69 6E 74 2B 92 01 04 :38
4F90 6C 69 6E 65 2B D0 01 04 68 65 72 65 2B D6 19 04 :6A
4FA0 6D 6F 64 65 2B DE 01 05 61 6C 6C 6F 74 2B E7 01 :E3
4FB0 01 2C 2B F2 01 02 63 2C 2B FD 19 01 27 2C 07 01 :79
4FC0 07 69 6E 69 74 69 61 6C 2C 0D 01 09 75 6E 64 65 :E0
4FD0 66 69 6E 65 64 2C 13 01 03 61 6E 64 2C 25 01 02 :D0
4FE0 6F 72 2C 2E 01 01 3C 2C 3F 01 01 3E 2C 50 80 06 :26
4FF0 3C 62 75 69 6C 64 2C 87 11 05 64 6F 65 73 3E 2C :2A
-------------------------------------------------------
Sum: A3 35 60 1F D8 42 D8 98 40 0E 19 66 A3 63 3D 2F :20
```

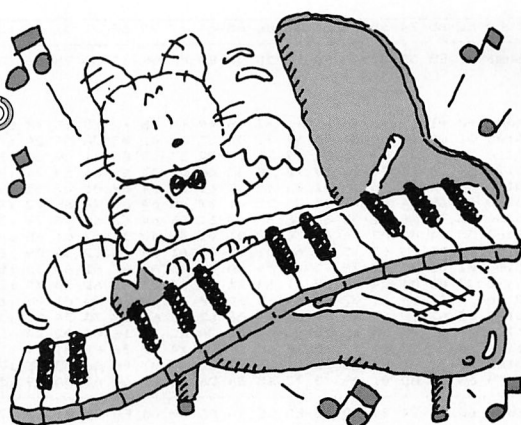
```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5000 D1 01 04 73 6E 6F 63 2C D7 01 08 66 6C 61 67 74 :A3
5010 65 73 74 2C F0 01 04 64 72 61 77 2D 1F 01 06 75 :E3
5020 6E 64 72 61 77 2D 24 01 0D 62 6C 6F 63 6B 74 72 :6C
5030 61 6E 73 66 65 72 25 16 01 02 2A 70 25 16 01 02 :95
5040 2A 71 25 16 01 01 6E 2D 8E 01 07 62 75 69 6C 74 :29
5050 69 6E 25 16 01 01 78 25 16 01 01 70 25 16 01 01 :76
5060 71 2D EF 01 04 73 61 76 65 2E 5E 01 05 68 65 6C :0C
5070 70 32 25 16 01 05 75 6E 64 65 66 2E F5 01 04 68 :85
5080 65 6C 70 2F 1E 01 05 65 63 68 6F 32 25 16 01 01 :A2
5090 79 25 16 01 01 7A 25 16 01 01 77 2F E5 01 12 69 :74
50A0 6E 73 65 72 74 5F 6F 72 5F 72 65 70 6C 61 63 65 :A7
50B0 32 2F 4B 01 11 69 6E 73 65 72 74 5F 6F 72 5F 72 :64
50C0 65 70 6C 61 63 65 25 16 01 05 25 16 01 01 6C 2E :B5
50D0 2F F8 01 06 64 75 6D 70 69 6E 65 30 30 01 0C 75 6E :0B
50E0 70 61 63 6B 6E 75 6D 62 65 72 30 72 01 06 64 65 :9A
```

リスト 2　Prolog/FV. 2 マシン語ダンプ・リスト

```
50F0 6C 65 74 65 30 CF 01 06 65 64 69 74 6F 72 31 D0 :38
--------------------------------------------------------
Sum: 67 E5 35 83 4A EA 73 2B 20 ED CE DE 13 3A 98 F6 :6A


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5100 01 07 65 64 69 74 6F 72 32 25 16 01 03 2A 54 2A :A8
5110 32 44 01 07 65 64 69 74 6F 72 33 32 E0 01 09 69 :BD
5120 6E 70 75 74 6C 69 73 74 33 4E 01 07 65 64 69 74 :B2
5130 6F 72 34 25 16 01 04 66 6C 61 67 34 09 01 07 65 :99
5140 64 69 74 6F 72 35 34 2C 01 07 65 64 69 74 6F 72 :46
5150 36 34 A6 01 0B 63 6F 6D 6D 61 6E 64 6C 69 73 74 :B7
5160 35 24 01 02 74 6F 3E BD 01 02 40 40 35 57 01 05 :4F
5170 61 73 73 6F 63 35 A0 01 05 73 61 76 65 32 36 59 :64
5180 01 05 73 61 76 65 31 36 9A 01 05 6C 6F 61 64 32 :8E
5190 37 08 01 04 6C 6F 61 64 25 16 01 05 65 6D 70 74 :DB
51A0 79 37 D9 01 07 65 64 69 74 6F 72 70 38 9A 01 05 :60
51B0 6E 63 6F 6E 63 37 2E 01 06 65 6E 6C 69 73 74 25 :31
51C0 16 01 02 2A 65 25 16 01 02 2A 6C 25 16 01 02 2A :E4
51D0 6E 38 D7 01 07 65 6E 6C 69 73 74 32 3B EA 01 04 :70
51E0 70 72 65 64 28 CB 01 01 5E 3B F7 01 04 66 75 6E :7E
51F0 63 39 8D 01 09 70 72 65 64 26 66 75 6E 63 3A 69 :53
--------------------------------------------------------
Sum: B6 EC 24 49 8D B3 EB EE 1A 0C 48 06 F8 85 E1 85 :7F


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5200 01 07 65 6E 6C 69 73 74 34 3B 1A 01 07 65 6E 6C :67
5210 69 73 74 33 41 E3 01 04 65 78 65 63 3B BC 01 08 :51
5220 65 64 69 74 6F 72 70 32 3C 3D 01 05 65 78 65 63 :4D
5230 32 25 16 01 05 65 78 65 63 6E 25 16 01 02 2A 78 :66
5240 25 16 01 02 2A 79 3C 2B 01 03 76 61 72 3F B8 01 :8D
5250 06 70 72 6F 6C 6F 67 25 16 01 01 66 25 16 01 02 :7A
5260 70 65 25 16 01 02 66 65 25 16 01 05 65 72 72 6F :D7
5270 72 3C 04 01 05 70 72 65 64 32 3C 11 01 05 66 75 :C3
5280 6E 63 32 3C 1E 01 02 78 78 3C 8C 01 0A 70 72 65 :6A
5290 64 26 66 75 6E 63 70 3C BC 01 0A 70 72 65 64 26 :7A
52A0 66 75 6E 63 66 42 A2 01 05 70 72 69 6E 74 3C EC :51
52B0 01 06 70 72 69 6E 74 32 41 BD 01 05 66 65 74 63 :0C
52C0 68 3D 78 01 06 70 72 69 6E 74 33 3D AD 01 06 70 :E5
52D0 72 69 6E 74 34 3D EF 01 06 70 72 69 6E 74 35 42 :C8
52E0 DE 01 05 75 6E 69 66 79 44 09 01 04 6C 69 6E 6B :0F
52F0 43 6D 01 06 75 6E 69 66 79 32 F7 0D 01 01 61 25 :A0
--------------------------------------------------------
Sum: 42 42 56 14 35 15 8F 59 83 33 FF F2 7D F4 1F 52 :A9


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5300 16 01 01 62 25 16 01 01 63 46 38 01 06 72 65 66 :DC
5310 75 74 65 25 16 01 03 63 75 65 47 14 01 07 72 65 :04
5320 73 6F 6C 76 65 45 EB 01 04 75 6E 64 6F 47 5D 01 :B9
5330 08 72 65 73 6F 6C 76 65 32 25 16 01 06 61 70 70 :BD
5340 65 6E 64 25 16 01 02 2A 7A 25 16 01 02 2A 61 25 :07
5350 16 01 02 2A 62 48 44 01 06 74 72 79 73 79 73 25 :1B
5360 16 01 01 6D 25 16 01 01 64 44 DE 01 04 76 61 72 :96
5370 3F 25 16 01 02 65 78 49 1C 01 07 74 72 79 73 79 :12
5380 73 32 3E B9 01 02 64 6F 3E FA 01 05 74 72 61 63 :5A
5390 65 25 16 01 01 72 40 AC 01 09 74 65 72 6D 5F 65 :86
53A0 64 69 74 25 16 01 04 65 64 69 74 25 16 01 0C 70 :DF
53B0 72 65 64 5F 6F 72 5F 66 75 6E 63 3F 07 01 0A 70 :47
53C0 72 6F 6C 6F 67 69 6E 69 74 41 60 01 07 70 72 6F :D1
53D0 6C 6F 61 64 25 16 01 07 70 72 65 64 5F 65 64 25 :DB
53E0 16 01 07 66 75 6E 63 5F 65 64 3E ED 01 08 65 78 :03
53F0 65 63 6C 69 73 74 4D 0C 01 07 70 72 6F 6C 6F 67 :78
--------------------------------------------------------
Sum: DD 52 20 0D A9 D4 4A 00 70 1B 2F FB 40 DD CC 8C :4D


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5400 32 25 16 01 0D 21 21 21 73 75 63 63 65 73 73 21 :F8
5410 21 21 25 16 01 0D 21 21 21 66 61 69 6C 75 72 65 :D6
5420 21 21 21 4B 54 01 06 75 6E 69 66 79 33 45 3E 01 :EB
5430 02 74 72 25 16 01 0A 2A 2A 49 73 54 72 75 65 0D :EB
5440 0A 25 16 01 06 2A 2A 2A 49 66 3A 25 16 01 04 2C :1F
5450 61 6E 64 49 BF 01 07 74 72 79 73 79 73 33 F1 A8 :CD
5460 01 09 75 2D 61 64 64 72 65 73 73 44 D8 01 0A 75 :2E
5470 5F 73 74 61 63 6B 5F 61 74 44 DB 01 0B 75 5F 73 :1B
5480 74 61 63 6B 5F 73 65 74 25 16 01 01 75 45 13 01 :59
5490 04 61 74 6F 6D 4A 02 01 07 74 72 79 73 79 73 34 :FB
54A0 4B 2B 01 06 66 65 74 63 68 32 4A FB 01 07 74 72 :EC
54B0 79 73 79 73 35 4C 17 01 06 75 6E 69 66 79 34 25 :FB
54C0 16 01 01 76 4C F5 01 02 74 3F 25 16 01 04 66 72 :9D
54D0 6F 6D 4D B5 01 03 6D 6F 64 25 16 01 06 66 69 6C :9F
54E0 74 65 72 25 16 01 05 73 69 65 76 65 25 16 01 09 :ED
54F0 70 72 69 6E 74 5F 61 6C 6C 25 16 01 03 BF BD B3 :33
--------------------------------------------------------
Sum: E6 8F AB 70 3F F0 0C 7B 07 42 8A D7 60 C9 A1 B6 :70


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5500 25 16 01 09 61 6C 6C 5F 70 72 69 6E 74 1E 2D 01 :56
5510 05 62 61 73 69 63 25 16 01 03 6D 26 63 25 16 01 :78
5520 07 2A 6D 65 6D 6F 72 79 25 16 01 03 2A 72 6D 25 :37
5530 16 01 03 2A 72 63 25 16 01 03 2A 6C 6D 25 16 01 :97
5540 03 2A 6C 63 25 16 01 02 2A 72 25 16 01 08 74 72 :00
5550 61 6E 73 66 65 72 25 16 01 03 2A 62 27 25 16 01 :AD
5560 03 2A 72 27 25 16 01 03 2A 6C 27 25 16 01 04 73 :75
5570 61 66 65 25 16 01 06 6D 65 6D 6F 72 79 25 16 01 :43
5580 08 2A 6D 65 6D 6F 72 79 27 25 16 01 05 74 72 61 :7A
5590 6E 73 25 16 01 02 2A 63 25 16 01 02 2A 6D 25 16 :BC
55A0 01 07 67 72 65 61 74 65 72 25 16 01 07 6E 6F 74 :86
55B0 6C 65 73 73 25 16 01 07 69 6E 63 6C 75 64 65 25 :03
55C0 16 01 04 B9 B2 DA B7 25 16 01 06 70 75 7A 7A 6C :9E
55D0 65 25 16 01 02 BC C0 25 16 01 02 B3 B4 25 16 01 :00
55E0 04 CB C0 DE D8 25 16 01 03 D0 B7 DE 25 16 01 0B :30
55F0 6E 75 6D 62 65 72 5F 6C 69 73 74 25 16 01 05 68 :4D
--------------------------------------------------------
Sum: DF 3A 3B 7A 57 55 52 8B 10 EF A9 A8 34 96 6B FF :DB


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5600 61 6E 6F 69 25 16 01 01 A6 25 16 01 02 B6 D7 25 :7A
5610 16 01 05 CD B3 C2 BD A1 25 16 01 06 6D 65 6D 62 :9F
5620 65 72 25 16 01 09 6D 65 6D 62 65 72 61 6C 6C 25 :F2
5630 16 01 04 74 72 6F 6E 25 16 01 07 27 72 65 66 75 :FA
5640 74 65 25 16 01 02 85 3B 0C 00 00 00 00 00 00 00 :E3
--------------------------------------------------------
Sum: 66 47 C2 D6 4C 52 1E 67 5A 9E 83 A0 42 EC 16 21 :E8
```

FM-7

ミュージック・メーカー

MUSIC MAKER Ver.1.0

■COMPAC T

現在，PSGが載っているパソコンでは，BASICのPLAY文などで簡単に音が出せます．しかし，マシン語で同程度に音を出そうとすると，専用のコードに従ってデータを新たに作らなければならないなど，意外に面倒でした．そこで発想を変えて，鋭く迫ってみました．あなたのゲームに付けて，もっとゲームを楽しくしてみませんか．

特徴

❶データ作りがきわめて簡単．面倒な操作は一切なし．
❷256バイト単位(××00)で，リロケータブル．
❸タイマ割り込みを使っているので，メイン側のプログラムと同時進行可能．
❹オールRAM動作可能．

原理

BASICではPLAY文がくると，タイマ割り込みによって順々にPSGへデータを送り，音を出します．本プログラムではここに着目し，PSGに送られたデータをPSGから直接読んで，次々にデータとして落してゆきます．このため，面倒な手間を必要とせず，バカチョン式にデータができます．

使い方

リスト1はPSGを読んで，データを作るプログラムで，**ジェネレータ**と言い，**リスト2**はできたデータを演奏するプログラムで，**シーケンサ**といいます．それぞれ，SAVEM "MM. G", &H4000, &H418C, &H4000とSAVEM "MM. S", &H4200, &H42E4, &H4200でセーブしてください．

使い方は非常に簡単．BASICの**PLAY**文で，音楽演奏のルーチンを作ったら，このジェネレータをロードし，**PLAY**文の前に**EXEC &H4000**と入れて，ただRUNするだけです．音楽が終わってReadyの表示が出たら，BREAK キーを押してください．画面に"CODE TOP"と"CODE END"のアドレスが表示されるので，セーブするときはこの値に従ってセーブします．なお，ジェネレータをロードする前に，必ず**CLEAR, &H3FFF**をしてください．もし"Out of memory" の表が出たら，BASICが大きすぎるので，適当にけずってください．このままでも，12Kくらいのデータをジェネレートできます．

例1　ジェネレート例

```
PLAY "T60S1M5000L8CDE"

4300:07 F8                SOUND7,&HF8
4302:0D 01                S1
4304:0C 13 0B 88          M5000
4308:08 11                CH-A ｴﾝﾍﾞﾛｰﾌﾟ
430A:01 01 00 25          C
430E:80 FA                (T60L8) ﾉ ｼﾞｶﾝ
4310:00 05                D
4312:80 FA
4314:01 00 00 E9          E
4318:80 FA
431A:0D 00                PSG ﾉ ﾘｾｯﾄ
431C:0C 00 0B 00
4320:08 00
4322:00 00
4324:FF FF                END ｺｰﾄﾞ
```

CONTINUE ?について

いくつものデータを作るとき，この"CONTINUE?"に対して N 以外のキーを押すと，以前に作ったデータの後から，新しいデータをジェネレートしていきます．N を押すと，ポインタを最初の値 ($4300) に戻し，そこからジェネレートします．

ポインタについて

ジェネレートされるデータのポインタは，**リスト1**の&H4004番地に，$4300として入っています．この&H4004番地を書き換えると，そのアドレスから，データを作っていきます．なお，この値はリロケートしても変わりません．

データについて

PSGのレジスタ1個につき，2バイトのペアとしてジェネレートされます．このうち，$8001から$FEFFまでの値はカウント数で，この値から$8000を引いて2倍した値が，msで表わされる待ち時間になります．$FFFFはデータの終了を示します．**例1**を参照してください．

シーケンサについて

ジェネレータで作ったデータを演奏します．タイマ(20ms) のIRQ割り込みを使う点は，BASICの**PLAY**文と同じで

例2　ジェネレートのメモリ節約

```
①   100 FOR I=0 TO 5
     110 PLAY "CEG"
     120 NEXT

②   100 EXEC &H4000
     110 PLAY "CEG"
     120 END
```

す．したがって，メインとなるプログラム（ゲームなど）と，同時進行します．ここで，このシーケンサの持つコードについて説明します．

LOOP（$0E*nn*）について：PSGのレジスタは14個で，$00～$0Dまでなので，あると便利なループを作りました．*nn*の値によって，LOOP-POOL間を繰り返します．*nn*＝0とすると，無限ループになります．なお，ネスティングは何重でも可能ですが，シーケンサのアドレス（$4200）から$0000に向けて，3バイトずつ使います．

POOL（$0F××）について：上記のLOOPに対応します．××の値は何でもかまいませんが，2バイトいります．BASICの**FOR～NEXT**と同じですが，POOLは最も近いLOOPと対応します．

上記の2つのコードは，ジェネレータにはありません．また，ジェネレータは，**SOUND**文をジェネレートすることはできません．しかし，これらは先に説明したジェネレータのポインタの書き換え（&H4004）と組み合せて，容易に手入力することができます．**例2**を見てください．

①をそのままジェネレートすると，コードが多くなってしまいます．そこで，②のように書き換え，モニタからジェネレータのポインタを2バイトずらします（&H4004，5のデータ$4300 ⇒ $4302）．そして，空いた2バイトに，**LOOP**コードと回数（&H4300，1に$0E05）を書き込みます．BASICに戻り，ジェネレートした後，コード・エンドのアドレスに，**POOL**コードとエンド・コードを書き足します．

```
CODE END=$431B・・・・・・・・・・・・・・・&H431B, C : $FFFF
    &H431B, C : $0F00
    &H431D, E : $FFFF
```

そしてポインタを，コード・エンド＋2バイトに書き換えます．

```
&H4004 : $4302 ⟶ $431E ($431B+2)
```

以上ですが，セーブするときは$4300～$431Eまでとなり，画面に表示されたものではなくなるので，注意してください．

つぎに，**SOUND**文をデータに変換する場合ですが，これは，BASICの**SOUND**文と1対1で対応しているので，モニタから次々に16進で書き込んでください．例として，FM-7の〝ユーザーズマニュアル　システム解説8-8〞に載っている，『波の音』をデータにしてみます．

アドレス	データ	コメント
4300	07 F7	OSC. set
4302	06 45	Noise FRQ. set
4304	08 10	Set Envelope. <part a>
4306	0B 00	Envelope FRQ. set
4308	0C 37	
430A	0D 0E	Set envelope wave pattern.
430C	93 88	Wait数（9388_{16}-8000_{16}）×2/1000＝10秒
430E	FF FF	エンド・コード
		注）マニュアル中は10，16進数ですが，ここではすべて16進数

Wait数は，次のコード（$430E～）を実行するまでのカウンタで，先にも説明した通り，$8001(20ms)～$FEFF(65秒)までです．

シーケンサの使い方

コードが（シーケンサのアドレス＋$100）からのときは，シーケンサのトップ・アドレスがスタートで，それ以外のときはXレジスタにコードのアドレスを入れて，（トップ・アドレス＋2）を呼びます．

オールRAM動作について

シーケンサは，システムの割り込みベクタ（$FFF8番地）を書き換え，タイマ割り込みがきたとき，まず自分のルーチンを通したあと，システムの割り込みサービス・ルーチンにジャンプしていくようになっています．割り込みフラグ（$FD03）は，1度読むと全部のフラグが落されてしまうので，本プログラム内では読んでいません．したがって，オールRAM動作時にはIRQのサービス・ルーチンを持たない場合は，ジャンプ先でフラグを落さないと，IRQが掛りっぱなしになってしまうので，注意してください．$FFF8のベクタが$01DDのとき，下記のようにします．

```
$01DD : 7D FD03     TST $FD03
$01E0 : 3B          RTI
```

リスト1　（ジェネレータ）

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
4000	20	49	01	DD	43	00	00	00	00	00	00	00	00	00	00	00	:8A
4010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:00
4020	00	00	00	00	00	00	00	0D	0A	43	4F	44	45	20	54	4F	:F5
4030	50	20	3D	24	43	4F	44	45	20	45	4E	44	20	3D	24	43	:A7
4040	4F	4E	54	49	4E	55	45	20	3F	0D	0A	34	1E	1A	50	1F	:73
4050	50	1F	8B	0D	08	26	08	0F	08	DC	04	DD	06	20	2D	31	:95
4060	8C	DD	C6	0C	17	00	E5	30	8C	B1	31	8C	B4	86	15	A7	:57
4070	84	10	AF	02	AD	9F	FB	FA	6D	21	27	F8	A6	A4	81	4E	:4C
4080	27	D5	81	6E	27	D1	9E	06	30	01	9F	06	31	8C	9A	C6	:7A
4090	0B	17	00	B8	17	00	C8	9E	06	CC	07	F8	ED	81	9F	06	:3B
40A0	86	11	30	8D	FF	63	6F	86	4A	2A	FB	30	8D	FF	5E	86	:BA
40B0	0D	8D	0F	E7	86	4A	2A	F9	30	8C	21	BF	FF	F8	1C	AF	:E1
40C0	35	9E	34	02	B7	FD	0E	86	03	B7	FD	0D	7F	FD	0D	86	:24
40D0	01	B7	FD	0D	F6	FD	0E	7F	FD	0D	35	82	1F	50	1F	8B	:1C
40E0	9E	0B	30	01	9F	0B	9E	06	31	8D	FF	21	86	0D	8D	D2	:F8

リスト1 （ジェネレータ）

```
40F0 E1 A6 27 22 0D 09 26 04 03 09 26 0E 0D 0A 26 10 :9D
-----------------------------------------------------------
Sum: 99 53 DA 31 BC F5 50 DD 4E 20 1C C8 BE 29 1D CB :F6


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4100 34 06 DC 0B 8A 80 ED 81 35 06 03 0A 0F 0B 0F 0C :16
4110 E7 A6 ED 81 9F 06 4A 2A D5 0F 0A 0D 09 27 29 7D :E5
4120 05 C0 27 05 8C 6F F0 25 1F CC FF FF ED 80 9F 06 :FC
4130 7F FD 02 31 8D FE FD C6 0B 8D 11 8D 22 0D 08 26 :90
4140 02 03 08 DC 02 FD FF F8 9E 02 6E 84 30 8D FE CB :F7
4150 10 AF 02 4F ED 04 86 14 A7 84 AD 9F FB FA 39 30 :70
4160 8D FE A3 31 8D FE BC 8D 0B 8D 09 31 8D FE B4 C6 :0A
4170 06 8D D9 39 A6 84 C6 10 3D 8D 07 A6 80 84 0F 8D :BC
4180 01 39 8B 30 81 39 23 02 8B 07 A7 A0 39 00 00 00 :E6
4190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
41A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
41B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
41C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
41D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
41E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
41F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-----------------------------------------------------------
Sum: 45 DF 03 87 E5 AF 4E 41 4C 15 EF 3D 98 C8 D9 03 :9A
```

リスト2 （シーケンサ）

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4200 20 14 20 0C 20 47 00 00 00 00 00 00 00 00 00 00 :C7
4210 34 1E 1A 50 20 0C 34 1E 1A 50 30 8C E3 CC 01 00 :10
4220 30 8B 1F 50 1F 8B 9F 0A 30 8C D5 9F 08 0F 0C 0F :DF
4230 0D 0F 0F 0D 0E 26 07 03 0E FC FF F8 DD 06 30 8C :16
4240 48 BF FF F8 86 04 B7 FD 02 1C AF 35 9E 34 06 1A :30
4250 50 8D 04 1C AF 35 86 CC 0D 00 8D 14 4A 2A FB CC :1C
4260 07 FF 8D 0C 7F FD 02 0F 0E EC 8C 9A FD FF F8 39 :79
4270 34 02 B7 FD 0E 86 03 B7 FD 0D 7F FD 0D F7 FD 0E :CD
4280 4A B7 FD 0D 7F FD 0D 35 82 1F 50 1F 8B 9E 0A DE :EA
4290 08 DC 0C 27 07 C3 FF FF DD 0C 20 43 EC 81 2B 32 :F5
42A0 81 0E 26 06 36 14 DF 08 20 F2 81 0F 26 10 6D C4 :F5
42B0 27 08 6A C4 26 04 33 43 20 EC AE 41 20 DE 81 0D :84
42C0 26 02 D7 0F 8D AA 81 06 22 D2 86 0D D6 0F 8D A0 :65
42D0 20 CA 81 FF 26 05 17 FF 7E 20 04 84 7F DD 0C 9F :D8
42E0 0A 9E 06 6E 84 00 00 00 00 00 00 00 00 00 00 00 :A0
42F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-----------------------------------------------------------
Sum: AE 2C A6 50 48 47 D2 3E B1 E8 74 46 CC 2E EF E8 :93
```

RANDOM BOX

システム開発用簡単エディタEDIT_S

■T.T.S.

このプログラムは6809開発用に短かくて，ある程度以上の機能を持ったエディタが必要になり，作成したものです．

ユーザーは1文字入出力，エディット先頭，終了アドレス，モニタの入ロアドレス，スタック(S,U)のセットなどのパラメータを(アドレス値として)与えるだけで，実行できます．

また，256バイトでリロケータブルです．

コマンド

I ⏎：現在ポイントのある行の上に，行を挿入します．何も入力せずに⏎キーを押すと，コマンド待ち(プロンプト〝]〞)に戻ります．
K⏎：現在のポインタのある行を削除します．
＋(－)××⏎：現在のポインタから＋(－)99までの範囲の行にポインタを移します．
P ××⏎：現在のポインタのある行から××数の行を出力します．
F str⏎：現在のポインタのある行から，ボトムまでの間の文字列strをサーチし，あればその行にポインタを移します．
C str1⏎：**str2**⏎：現在のポインタのある行のstr1をstr2の文字列に置き換えます．必ず同じ文字数にしてください．
T⏎：行ポインタをトップに移します．
B⏎：行ポインタをボトムに移します．
S ⏎：モニタに戻ります．
＊：××は2桁の数値，strは文字列を示します．

注) ①1行の入力文字数の制限は80文字までです．
②それぞれのコマンドについて，BEEP音がし，ERRを表示してエラーを知らせます．
③ファイルの入出力ルーチンは別に作ってください．このシステムはエディタだけです．
④プリンタ出力は1文字出力ルーチン内で行なえば簡単でしょう．
⑤エディット・エリアをオーバーするとエラーになり，メッセージを出力した後，コマンド待ちに戻ります．
⑥$22AC～$22FFの間は1行入力用バッファなので，ここには何もおかないようにしてください．

最後になりましたが，このソフトがシステム開発，ソフト開発の手助けになれば幸いです．

アドレス・パラメータのセットの仕方

アドレス	データ	アドレス	データ
$2000	COLDスタート	$2010～11	Uスタックのデータをセット
$2002	HOTスタート	例 $1F82から1文字入力(エコーバック付)， $1F80に1文字出力ルーチンがあるとき．	
$2004～5	一文字入力ルーチンのアドレスをセット	$2004～5	$1F82
$2006～7	一文字出力ルーチンのアドレスをセット	$2006～7	$1F80
$2008～9	エディット・エリアの先頭アドレスをセット	$2008～9	$2300　エディット・
$200A～B	エディット・エリアの終了アドレスをセット	$200A～B	$3FFF　エリア
$200C～D	モニタの復帰アドレスをセット	$200C～D	$ABF4(FM-7)，$82B8(FM-11) モニタ
$200E～F	Sスタックのデータをセット	$200E~F	$5FFF (Sスタック)
		$2010~11	$5000 (Uスタック)

図1　ファイル構造

```
| 00 | 行 | 00 | 行 | ……… | 行 | 00 | 00 |   (ELP : End Line Pointer)
          ↑                        ↑
          └─行エンド                └─ELPエンド・ポインタ
```

リスト1　EDIT_S

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2000 20 15 20 28 1F 82 1F 80 30 00 4F FF AB F4 5F FF :38
2010 50 00 02 30 01 30 40 AE 8C EE 30 01 AF 8C F4 AF :2A
2020 8C F3 AE 8C E3 6F 80 AC 8C E0 25 F9 30 8D 00 78 :F6
2030 17 00 95 17 00 9F AE 8C DA 17 00 8C 10 EE 8C CE :71
2040 EE 8C CD 86 5D AD 9C BE 17 00 97 A6 80 27 45 81 :F2
2050 53 26 03 6E 9C B6 81 49 10 27 00 CE 81 4B 10 27 :0E
2060 01 65 81 42 10 27 01 24 81 54 10 27 01 11 81 46 :6A
2070 10 27 01 1E 81 43 10 27 01 D7 81 2B 10 27 01 7C :89
2080 81 2D 10 27 01 76 81 50 10 27 01 70 30 8D 00 32 :C4
2090 8D 36 20 AF AE 8D FF 7B AC 8D FF 79 24 08 A6 80 :4A
20A0 26 FC AF 8D FF 6D 20 8B 45 44 49 54 5F 53 20 52 :BF
20B0 45 4C 20 31 2E 30 0D 0A 70 72 64 20 57 4F 4C 46 :F5
20C0 2E 00 45 52 52 21 07 00 A6 80 27 06 AD 9D FF 36 :11
20D0 20 F6 8D 01 39 86 0D AD 9D FF 2B 86 0A AD 9D FF :BD
20E0 25 39 30 8D 01 C6 5F AD 9D FF 19 81 0D 27 2A 81 :03
20F0 08 27 0B A7 80 5C C1 50 25 ED 35 10 20 8E 31 8D :91
-----------------------------------------------------------
Sum: 59 47 C3 6A 75 F6 9C C2 41 0C 19 C5 9A DB BF EB :E0


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2100 01 AA 36 20 AC C1 23 1A 30 1F 5A 86 20 AD 9D FE :42
2110 F5 86 08 AD 9D FE EF 20 CE 6F 80 6F 80 30 8D 01 :44
2120 8B 39 86 20 AD 9D FE DE 20 B8 8D A9 EC 8D FE E5 :FA
2130 AC 8D FE D6 25 12 30 8D 00 05 8D 8C 16 FF 4D 41 :C2
2140 52 41 20 4F 56 52 21 00 8D 98 5D 10 27 FE E4 5C :C2
2150 AE 8D FE C1 31 85 10 AF 8D FE BA A6 84 A7 A4 AC :D5
2160 8D FE B0 27 06 30 1F 31 3F 20 F0 5A 31 8D 01 3C :8C
2170 A6 A0 A7 80 5A 26 F9 6F 80 AF 8D FE 96 20 AB AE :1E
2180 8D FE 85 30 01 AF 8D FE 8A 16 FE A7 AE 8D FE 85 :7E
2190 20 F3 5A 10 27 FE F5 E7 8D FE 77 AE 8D FE 74 31 :5E
21A0 8D 01 09 31 21 E6 8D FE 69 AC 8D FE 68 10 24 FE :94
21B0 DB A6 80 A1 A0 26 E8 5A 26 F7 A6 82 26 FC 30 01 :42
21C0 AF 8D FE 4F 16 FE 6C AE 8D FE 48 A6 80 26 FC 10 :E2
21D0 AE 8D FE 3F A6 80 A7 A0 4D 26 F9 A6 84 26 F5 10 :A6
21E0 AF 8D FE 31 6F A0 6F A0 16 FE 48 A6 80 80 30 10 :CB
21F0 2B FF 07 81 0A 10 24 FF 01 84 0F 39 8D ED C6 0A :06
-----------------------------------------------------------
Sum: AC A0 A0 CC 20 82 26 1E 8E 0D C8 38 EE 0B 56 06 :8E


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2200 3D 8D E8 36 02 EB C0 AE 8D FE 08 A6 8D 00 9D 81 :27
2210 2B 27 16 81 2D 27 24 17 FE BB 17 FE AB AC 8D FD :27
2220 F4 27 03 5A 26 F4 16 FE 0A AC 8D FD E8 10 24 FF :01
2230 5B A6 80 26 FC 5A 26 F1 16 FF 4A 30 1F AC 8D FD :F8
2240 C7 10 23 FF 3A A6 82 26 FC 5A 26 F1 30 01 16 FF :34
2250 34 17 FE 81 5A 10 27 FE 33 E7 8D FD B5 AE 8D FD :EA
2260 B2 31 8D 00 47 31 21 E6 8D FD A7 A6 80 10 27 FE :7B
2270 1B A1 A0 26 EC 5A 26 F3 E6 8D FD 96 30 1F 5A 26 :B6
2280 FB 36 10 86 20 AD 9D FD 7D 86 3A AD 9D FD 77 17 :40
2290 FE 50 E1 8D FD 7C 10 26 FD F2 37 10 31 8D 00 0C :6B
22A0 A6 A0 A7 80 5A 26 F9 16 FD 89 00 00 53 00 00 00 :D5
-----------------------------------------------------------
Sum: 1E A0 67 70 8F F0 B6 EA C4 30 BE B8 F5 D0 76 BD :16
```

リスト2　1文字入出力ルーチン(エンジン・ルーム作)

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1F80 20 3D 34 04 BD 1F C6 CC 29 01 FD FC 82 BD 1F D3 :57
1F90 BD 1F C6 C6 80 B6 FC 83 97 C5 FA FC 80 F7 FC 80 :62
1FA0 BD 1F D3 BD 1F C6 CC 03 01 FD FC 82 96 C5 B7 FC :AA
1FB0 84 F6 FC 80 F7 1E 01 BD 1F D3 96 C5 35 04 39 34 :BC
1FC0 04 97 C5 20 DE 00 F6 FD 05 2B FB 1A 50 C6 80 F7 :23
1FD0 FD 05 39 7F FD 05 F6 FD 05 2B F8 39 00 00 00 00 :10
-----------------------------------------------------------
Sum: 1F 0D C7 A6 2E BE 7B 09 EA EC 7C 92 1D 43 8B 7A :52
```

ゲーム編

FM-7 Newゴルゴ13……ワンダーソフト Sgn 162
FM-7 FLYING SCISSORS……高須晶英 167
FM-7 INDY-7……高畑英樹 171
MELON PANIC(少しの改造で11対応)……三浦 貢 175
FM-7 詰め将棋……土肥一夫 179
FM-7 3D UFO……高畑英樹 190
将棋対局……MAX 193
FM-7 OCTPUS GARDEN……Open-Reach 200
FM-7 バトミントン・ゲーム……吉田 昌 205

画面写真は合成してあります。

> Kコンパイラで作った高速・音楽付ゲームです．サブシステムにパターンを送って表示(バドミントン・ゲーム参照)しています．音楽はメイン・システムで，IRQ時計割り込み利用のPSGサウンドです．ゴルゴ13の気分になって，名射撃を楽しんでください．
> また，このゲームはいろんな要素の入ったゲームなので，解析すれば各種のゲームが作れるでしょう．

物　　語

ゴルゴ13は，ドーソン会長が待ちうけるドーソン・ビルに単身でのりこみました．もちろん，ゴルゴ13はあなたです．ゴルゴ13が今回引き受けた依頼は，ビルの最上階まで行って，ドーソン会長と対決することです．でも，先には様々の障害が待ちうけています．

ドーソン会長は，とりわけ腕ききの殺し屋を3人雇っています．ヘビのようなスネーク，全身銀と金でおおわれた，不死身のシルバーとゴールドが待ちかまえています．スネーク，シルバー，ゴールドはブロックを投げつつ，ゴルゴ13に迫ってきます．ブロックはときには，積み重なり，ときにはゴルゴ13を直撃します．

ゴルゴ13の銃弾は，ブロックを壊したり，壁やブロックにはね返りながら相手をねらいます．39階では恐怖の停電があり，相手の姿が見えません．この階をクリアすればいよいよ最上階で，ドーソン会長が一人で待ちうけています．

ゲームの方法

ゴルゴ13を左右に動かして，相手を攻撃します．キー操作は**表1**，得点は**表2**の通りです．ゴルゴ13は影武者を含めて4人います．全員やられるとゲーム終了です．40階をクリアすれば，ゴルゴ13が一人追加されます．

プログラムの作り方

①Kコンパイラとランタイム・ルーチンをロードし，裏RAMの$C000以降にリロケータでリロケートします．
②**リスト1**を全部打ち込みます．文番号の後には，必らずCLEAR 50, &H6FFF [RETURN] をした後，“,” をつけてから打ち込んでください．
③エグゼキュータをロードして，$C000から実行させると，コンパイルを始めます．
④コンパイル・エラーがあったときは，それを修正して③を繰り返してください．
⑤エラーがなくなったら，プログラムをファイルにセーブしておきます．
⑥次に，MONコマンドでモニタ・モードにして，Mコマンドで**リスト2**のデータを打ち込みます．必ずチェック・サムを確認しておいてください．
⑦全部できたら，SAVEMコマンドで，$3C00から$6EFFまでを，スタート番地を$5000にしてセーブします．
⑧EXEC & H5000でゲームが始まります．
バグがあるときは，④に戻って，繰り返してください．

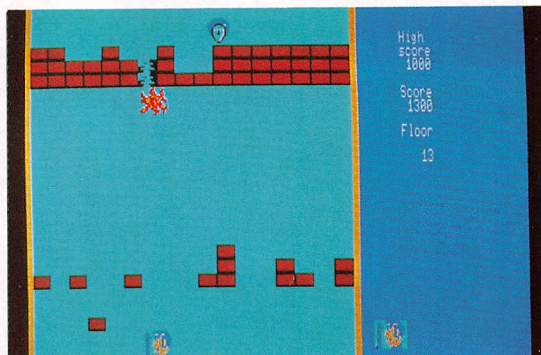



写真1　やっと次に進めるゾ

プログラムの説明

『Kコンパイラ有象無象』で紹介したテクニックを，いろんなところで使っています．そちらを参照してください．

音楽および音は，『ミュージック・メーカー』を利用しています．このルーチンは，IRQを使った時間割り込みを使っているので，音楽や音を出しても画面の方には影響なくゲームが進みます．

写真提供：東宝東和「ゴルゴ13」より

表1　各キーの説明

キー	説明
1	ゴルゴが左に移動.
2	ゴルゴが立ち止まる.
3	ゴルゴが右に移動.
4	銃を左上に向ける. 弾はブロック, 壁ではね返る.
5	銃を真上に向ける. 弾はブロックを壊す.
6	銃を右上に向ける. 弾はブロック, 壁ではね返る.
PF1	銃弾を撃つ.

表2　得点表

スネーク	100点	1階から39階まで登場 } 一人でも倒せば上の階に上がれる.
シルバー	200点	11階から39階まで登場
ゴールド	300点	21階から39階まで登場
ドーソン	500点	40階にのみ登場.
返射銃弾	10点	銃弾が壁やブロックにはね返って敵に当ると, はね返った回数だけ倍増されて, 加算される.

表3　パターン・データ・エリア(先頭$4100)

アドレス	シンボル	サイズ	内容
100	PDEL	5×16×3	消去用
1F0	PGBR	5×16×3	ゴルゴ右向
2E0	PGBD	5×16×3	ゴルゴ左向
3D0	PGBL	1×4×3	銃弾(ゴルゴ)
3DC	PSND	1×4×3	銃弾
3E8	PBLK	3×8×3	ブロック
430	PSNK	4×16×3	スネーク
4F0	PGLD	4×16×3	ゴールド
5B0	PSLV	4×16×3	シルバー
670	PDSN	4×16×3	ドーソン
730	PHUP	1×8×3	上向き炎
748	PHDW	3×8×3	崩れブロック
790	PGBB	5×16×3	爆発パターン1
880	PSNB	5×16×3	爆発パターン2
970	PSDB	1×16×3	弾衝突パターン

リスト1　Newゴルゴ13 ソース・リスト

```
10 'CONST PDEL=$100,PGBR=$1F0,PGBL=$2E0,PGBD=$3D0,PSND=$3DC,PBLK=$3E8,PSNK=$430,
PGLD=$4F0,PSLV=$5B0,PDSN=$670,PHUP=$730,PHDW=$748,PGBB=$790,PSNB=$880,PSDB=$970
20 'CONST DATA=$2000,GY=23,BACK=5,SUBDATA=$4100,HRX=468,SSUB=$3C02,SSTOP=$3C04,S
HASHA=$3F0A,SBLOCK=$3F6F,SBACK=$3E55,STHEMA=$3D00,SREF=$3F7E,ISAVE=$3F8A,PFBIT=$
1
30 'CODE $10CE#,$1FFF#;INIG[]
40 'IRQA=$1DD;POKE IRQA,$7E;IRQA=IRQA+1;IRQB=IRQA(0);A=ISAVE;IRQA(0)=A;CPOKE A,$
7D,$FD03#,$3B
50 'HSC=0;GY8=GY*8;GYY=GY8-8
60 'GA=DATA;WORK=GA+1300;ASC=WORK+10;TEMP=ASC+4;TRANSFER[];DSCR1[HRX,32]
70 'INIT;SC=0;NGB=4;DSCR1[HRX,56]
80 'INIT1;A=NGB-2;IF A>3 THEN A=3;FI
90 'FOR I=0 TO A;FPUT[55+I*6,179,5,16,PGBL,8];NEXT
100 'FLR=0;TEMP:9)=-1;TEMP(5)=-1
110 'INIT2;ESST[]
120 'FLR=FLR+1;IF FLR>40 THEN NGB=NGB+1;FLASH[1,5];GOTO INIT1;FI
130 'ASC:0)=10;ASC:1)=20;ASC:2)=30
140 'FOR I=0 TO 7;TEMP:I)=0;NEXT;A=EVEN[FLR]
150 'IF FLR<>40 THEN IF FLR>=1 THEN TEMP:0)=1;IF A THEN TEMP:9)=TEMP:9)+1;FI;TEM
P(11)=PSNK;FI;IF FLR>=11 THEN TEMP:1)=1;IF A THEN TEMP:10)=TEMP:10)+1;FI;TEMP(12
)=PSLV;FI
160 'IF FLR>=21 THEN TEMP:2)=1;IF A THEN TEMP:11)=TEMP:11)+1;FI;TEMP(13)=PGLD;FI
;ELSE TEMP:2)=1;TEMP:11)=0;TEMP(13)=PDSN;ASC:3)=50;FI
170 'DSCR[HRX+8,84,FLR]
180 'LINE[0,0,423,199,0,5,2]
190 'LINE[0,0,7,199,0,6,2]
200 'LINE[416,0,423,199,0,6,2]
210 'BLOCK[];CK=0;GX=25;DP=PGBL;LR=0
220 'INIT3;NDAN=0;NBK=1;PF=0;WTIME=180;IF PEEK($FD00) AND PFBIT THEN WTIME=300;F
I
230 'FPUT[GX,GY8,5,16,DP,8];NDD=0;VVX=0;TEMP(9)=0;TEMP:20)=0
240 'LOOP;CK=CK+1;CKK=CK-CK/3*3
250 'IF CK>=WTIME OR PF<>0 THEN WTIME=0;IF FLR=40 THEN CKK=2;FI;SMOVE[CKK];FI
260 'KEYIN[];GDAN[]
270 'IF NDD=1 THEN GOTO GEND;FI
280 'IF NBK=0 THEN GOTO FEND;FI
290 'IF EVEN[CK] THEN SDAN[0];SDAN[1];SDAN[2];FI
300 'IF NDD=0 THEN  GOTO LOOP;FI
310 'GEND;IF NGB<>0 THEN FOR I=0 TO 2;IF TEMP:I) THEN SSET[TEMP:I+3),TEMP:I+9),0
];FI;NEXT;GOTO INIT3;FI
320 'IF SC>HSC THEN HSC=SC;DSCR[HRX,32,HSC];FI;
330 'SYMBOL[120,160,1,1,2,9,"GAME OVER"];SYMBOL[120,168,1,1,2,16,"GAME AGAIN(Y/N
)?"]
340 'REPEAT;KY=INKEY[];IF KY='Y' OR KY='y' THEN GOTO INIT;FI;UNTIL KY='N' OR KY=
'n'
350 'PRINT CHR$(12);ESTOP[];IRQA(0)=IRQB;END
360 'FEND;FLASH[2,2];GOTO INIT2
370 'FLASH;FOR I=0 TO %1;SCREEN[7,%2];WAIT[10];SCREEN[7,7];WAIT[10];NEXT;RETURN
380 'BLOCK;FOR I=0 TO 1299;GA:I)=0;NEXT;RETURN
390 'KEYIN;
400 'N=INKEY[]-$30
410 'IF N=1 THEN FPUT[GX,GY8,5,16,PDEL,BACK];LR=0;DP=PGBL
420 '   IF GX>1 THEN GX=GX-1;ELSE GX=47;FI;GOTO KEYDSP
430 'ELSE IF N=3 THEN FPUT[GX,GY8,5,16,PDEL,BACK];LR=1;DP=PGBR
440 '   IF GX<47 THEN GX=GX+1;ELSE GX=1;FI;GOTO KEYDSP
450 'ELSE IF N>3 AND N<7 THEN VVX=N-5;WAIT[1]
460 'ELSE IF NDAN=0 THEN
470 '   IF PF THEN PF=0
480 '       VY=-1;NDAN=1;VX=VVX
490 '       IF LR>0 THEN GDX=4;ELSE GDX=0;FI
500 '       GDX=GX+GDX
510 '       FPUT[GDX,GYY,1,8,PHUP,8];ESGS[]
520 '       X=GDX;Y=GY-1;Y8=GYY;BND=0
530 '       FPUT[GDX,GYY,1,8,PDEL,BACK]
540 '       ELSE WAIT[1];FI;FI;FI;FI;FI
550 'RETURN
560 'KEYDSP;FPUT[GX,GY8,5,16,DP,8];RETURN
570 'GDAN;
580 'IF NDAN=0 THEN RETURN;FI
590 'FPUT[X,Y8,1,4,PDEL,BACK];XN=X+VX;YN=Y+VY
600 'GDAN1;XY=YN*52+XN
```

リスト1　Newゴルゴ13 ソース・ソフト

```
610 'IF YN>=23 THEN IF XN>=GX AND XN<=GX+4 THEN GOTO GDEAD;FI;FI
620 'IF YN>=25 OR YN<0 THEN NDAN=0;RETURN;FI
630 'IF XN<1 OR XN>51 THEN VX=-VX;XN=X;ESRF[];BND=BND+1;GOTO GDAN1;FI
640 '    A=SEX(GA:XY))
650 '    IF A=0 THEN GOTO DSP;FI
660 '    IF A<0 THEN SB=-A-1;TEMP:SB)=0;SX=TEMP:3+SB);SY=TEMP:9+SB);SY8=SY*8
670 '    FOR I=0 TO 5;FPUT[SX,SY8,5,16,PSNB,8];ESGD[];FPUT[SX,SY8,5,16,PGBB,8];WA
IT[1];NEXT;FPUT[SX,SY8,5,16,PDEL,BACK]
680 '    IF BND<>0 THEN A=BND-1;FOR I=1 TO A;BND=BND*2;NEXT;FI;A=ASC:SB)+BND;NUMB
ER[SX*8,SY8+16,1,1,7,3,A*10]
690 '        SC=SC+A;DSCR[HRX,56,SC];NBK=0;RETURN;FI
700 '    IF VX=0 THEN NDAN=0;BDEL[XN,YN];ESBL[];RETURN;FI
710 '    B=GA:XY-52*VY);ESRF[];BND=BND+1
720 '        IF (A=1 AND B=1) OR (A=3 AND B=3) THEN VX=-VX;XN=X;C=GA:XN+52*YN);IF
C=1 OR C=3 THEN VY=-VY;YN=Y;FI
730 '    ELSE XN=X+VX;VY=-VY;YN=Y;FI
740 'DSP;X=XN;Y=YN;Y8=Y*8;IF FLR<>39 THEN FPUT[X,Y8,1,4,PGBD,8];FI
750 'RETURN
760 'BDEL;AA=A-1;FPUT[%1-AA,%2*8,3,8,PHDW,BACK];XY=XY-AA
770 'GA:XY)=0;GA:XY+1)=0;GA:XY+2)=0;RETURN
780 'DSCR;LINE[%1,%2,%1+39,%2+7,0,1,2];NUMBER[%1,%2,1,1,7,5,%3];RETURN
790 'DSCR1;LINE[%1,%2,%1+47,%2+7,0,1,2];SYMBOL[%1+40,%2,1,1,7,1,"0"];RETURN
800 'SMOVE;IF TEMP:%1)=0 THEN RETURN;FI
810 'ADAN=TEMP:%1+18);SX=TEMP:%1+3);SVX=SEX(TEMP:%1+6));SY=TEMP:%1+9);SY8=SY*8;D
S=TEMP(%1+11);SB=-(%1+1);RND1=RND(2)
820 'SNX=SX+SVX;IF SX<>0 THEN FPUT[SX,SY8,4,16,PDEL,BACK];FI
830 'IF SNX>=1 AND SNX<=48 THEN IJ=SY*52
840 '    IF SVX>0 THEN A=IJ+SX;IF CHKBK[A+4] THEN SVX=-1;SNX=SX
850 '        ELSE GA:A)=0;GA:A+52)=0;GA:A+4)=SB;GA:A+56)=SB;FI
860 '    ELSE A=IJ+SNX;IF CHKBK[A] THEN SVX=1;SNX=SX
870 '        ELSE GA:A)=SB;GA:A+52)=SB;GA:A+4)=0;GA:A+56)=0;FI;FI
880 'ELSE SSET[SX,SY,0];RND1=1
890 'SVX=RND(2)*2-1;SY8=SY*8;IF SVX=-1 THEN SNX=48;ELSE SNX=1;FI;I=1;WHILE I<=48
;IF SCHK[SNX,SY] AND SCHK[SNX,SY+1] THEN SSET[SNX,SY,SB];GOTO SMOVE1;FI;I=I+1;WE
ND;RETURN;FI
900 'SMOVE1;SX=SNX
910 'IF ADAN=0 THEN IF RND1=1 THEN SX1=SX/3*3+1;SY2=SY+2;IF Y=SY2 THEN IF X>=SX1
 AND X<=SX1+2 THEN GOTO SADD;FI;FI
920 'IF SCRBK[SX1,SY2] THEN ADAN=1;SETBK[SX1,SY2];TEMP:%1+12)=SX1;TEMP:%1+15)=SY
2;ESRF[];FI;FI;FI
930 'SADD;IF FLR<>39 THEN FPUT[SX,SY8,4,16,DS,8];FI
940 'TEMP:%1+18)=ADAN;TEMP:%1+3)=SX;TEMP:%1+6)=SVX;TEMP:%1+9)=SY;TEMP(%1+11)=DS
950 'RETURN
960 'CHKBK;IF SEX(GA:%1))=0 AND SEX(GA:%1+52))=0 THEN %1=0 ELSE %1=1;FI;RETURN
970 'SSET;SWORK=%2*52+%1;FOR WW=0 TO 3;GA:SWORK+WW)=%3;GA:SWORK+52+WW)=%3;NEXT;R
ETURN
980 'SCHK;SWORK=%1+%2*52;WW=0;WHILE WW<=3;IF SEX(GA:SWORK+WW))=0 THEN %1=1;RETUR
N;FI;WW=WW+1;WEND;%1=0;RETURN
990 'SDAN;ADAN=TEMP:%1+18);IF ADAN=0 THEN RETURN;FI
1000 'AX=TEMP:%1+12);AY=TEMP:%1+15);AY8=AY*8
1010 'AYN=AY+1
1020 'IF NDAN=1 THEN IF Y=AYN THEN IF X>=AX AND X<=AX+2 THEN RETURN;FI;FI;FI
1030 'IF RND(2) THEN IF AYN=20 THEN IF AX<GX OR AX>GX+4 THEN GOTO SDAN1;FI;FI;FI
1040 'IF AYN>=23 THEN IF GX<=AX AND AX<=GX+4 THEN DELBK[AX,AY];GOTO GDEAD;FI;FI
1050 'IF AYN=25 THEN DELBK[AX,AY];GOTO SDAN1;FI
1060 'IF GA:AX+52*AY)=0 THEN GOTO SDAN1;FI
1070 'IF SCRBK[AX,AYN]=0 THEN GOTO SDAN1;FI
1080 'DELBK[AX,AY];AY=AYN;SETBK[AX,AY]
1090 'TEMP:%1+15)=AY
1100 'RETURN
1110 'SDAN1;TEMP:%1+18)=0;RETURN
1120 'SCRBK;%1=%1+%2*52;WW=0;WHILE WW<=2;IF GA:%1+WW)<>0 THEN %1=0;RETURN;FI;WW=
WW+1;WEND;%1=1;RETURN
1130 'DELBK;FPUT[%1,%2*8,3,8,PDEL,BACK]
1140 '%1=%1+%2*52;WW=0;WHILE WW<=2;GA:%1+WW)=0;WW=WW+1;WEND;RETURN
1150 'SETBK;FPUT[%1,%2*8,3,8,PBLK,8]
1160 '%1=%1+%2*52;FOR WW=0 TO 2;GA:%1+WW)=WW+1;NEXT;RETURN
1170 'GDEAD;FOR I=0 TO 5;FPUT[GX,GY8,5,16,PGBB,8];ESGD[];FPUT[GX,GY8,5,16,PSNB,2
];WAIT[1];NEXT
1180 'I=0;WHILE I<=2;IF TEMP:I)=1 THEN SX=TEMP:I+3);SY=TEMP:I+9);FPUT[SX,SY*8,4,
16,PDEL,BACK];SSET[SX,SY,0];FI;I=I+1;WEND
1190 'FOR I=0 TO 2;IF TEMP:I+18)=1 THEN DELBK[TEMP:I+12),TEMP:I+15)];FI;NEXT
1200 'IF NDAN=1 THEN FPUT[X,Y8,1,4,PDEL,BACK];FI
1210 'FPUT[GX,GY8,5,16,PDEL,BACK];NGB=NGB-1;NDD=1;IF NGB>0 AND NGB<5 THEN FPUT[4
9+NGB*6,179,5,16,PDEL,1];FI;RETURN
1220 '(* REFLECT SOUND *)
1230 'ESRF;BEEP[1];WAIT[1];BEEP[0];RETURN
1240 '(* TEHMA MUSIC *)
1250 'ESST;CODE $8E,STHEMA#,$BD,SSUB#;RETURN
1260 '(* GOLGO SHOOT SOUND *)
1270 'ESGS;CODE $8E,SHASHA#,$BD,SSUB#;RETURN
1280 '(* BLOCK BREAK SOUND *)
1290 'ESBL;CODE $8E,SBLOCK#,$BD,SSUB#;RETURN
1300 '(* SOUND STOP *)
1310 'ESTOP;CODE $BD,SSTOP#;RETURN
1320 'ESGD;SOUND[7,7]
1330 'SOUND[8,6]
1340 'SOUND[9,7];SOUND[10,8]
1350 'FOR T=1 TO 10
1360 'SOUND[6,T];WAIT[1];NEXT
1370 'SOUND[10,0];SOUND[8,0];SOUND[9,0];RETURN
1380 'SOUND;POKE$FD0E,%1;POKE$FD0D,3;POKE$FD0D,0;POKE$FD0E,%2;POKE$FD0D,2;POKE$F
D0D,0;RETURN
1390 'BEEP;IF %1=1 THEN POKE $FD03,$81;ELSE POKE $FD03,$80;FI;RETURN
1400 'EVEN;%1=%1-%1/2*2;RETURN
1410 'WAIT;CODE $3278#;%3=300;IF PEEK($FD00) AND PFBIT THEN %3=500;FI;%1=1;WHILE
```

リスト 1　Newゴルゴ13 ソース・ソフト

```
 %1<=%5;%2=0;WHILE %2<=%3;%2=%2+1;WEND;%1=%1+1;WEND;CODE $3268#;RETURN
1420 'SCREEN;DMY=COMPL[%2*16+%1];POKE $FD37,DMY;RETURN
1430 'COMPL;CODE $E663#,$53,$4F;RETURN
1440 'LINE;HLT[];CPOKE ADDR,$15,%6,%5,%1#,%2#,%3#,%4#,%7;SUB[];RETURN
1450 'BTOD;CODE $3278#;%1=1;%2=0;%3=10000;DMY=%6
1460 'WHILE %2<=4;%0=%5/%3;%5=%5-%0*%3;%0=%0+$30
1470 'IF %1=1 THEN IF %0=$30 THEN %0=$20;ELSE %1=0;FI;FI
1480 'DMY:%2)=%0;%3=%3/10;%2=%2+1;WEND;CODE $3268#;RETURN
1490 'NUMBER;BTOD[%7,WORK]
1500 'SYMBOL[%1,%2,%3,%4,%5,%6,WORK+5-%6];RETURN
1510 'SYMBOL;CODE $3278#;HLT[];CPOKE ADDR,$19,%9,0,0,%7,%8,%5#,%6#,%10;DMY=%11;%
1=0;REPEAT;ADDR:%1)=DMY:%1);%1=%1+1;UNTIL %1=%10;SUB[];CODE $3268#;RETURN
1520 'INKEY;A=PEEK($FD00) AND $80;IF A THEN PF=A;FI;KY=PEEK($FD01);RETURN
1530 'HLT;CODE $B6,$FD05#,$2BFB#,$1A40#,$8680#,$B7,$FD05#,$B6,$FD05#,$2AFB#;ADDR
=$FC82;RETURN
1540 'SUB;POKE $FD05,0;RETURN
1550 '(* ************  TRANSFER ********* *)
1560 'TRANSFER;B=SUBPRO[]-12;SUBMIT[B,0]
1570 'FOR I=0 TO 14;ADDRES=SUBDATA+$100*I;SUBMIT[ADDRES,I+1];NEXT;RETURN
1580 '(* ************  SUBMIT ********* *)
1590 'SUBMIT;W=$C000+$100*%2;ADDRES=%1;FOR W1=0 TO 3;HLT[];CPOKE ADDR,$3F,'Y','A
','M','A','U','C','H','I',$91,$D393#,W#,$40#,$90;FOR W2=0 TO $3F;ADDR:W2)=ADDRES
:W2);NEXT;SUB[];ADDRES=ADDRES+$40;W=W+$40;NEXT;RETURN
1600 '(* *********  F PUT ************ *)
1610 'FPUT;W=%1+80*%2;HLT[];CPOKE ADDR,$3F,'Y','A','M','A','U','C','H','I',$91,$
D396#,$C000#,$0007#,$93,$C00C#,$90,W#,%3,%4,%5#,%6;SUB[];RETURN
1620 '
1630 '(* *** FPUT PROGRAM TRANSPORT  *** *)
1640 'SUBPRO;CODE $308C#,$03,$1F10#,$39
1650 'CODE $34,$37,$B6,$D4
1660 'CODE $09,$CC,$C0,$00,$E3,$8C,$ED,$ED
1670 'CODE $8C,$ED,$1F,$01,$6F,$8C,$EC,$1A
1680 'CODE $51,$69,$8C,$E7,$A6,$8C,$E4,$81
1690 'CODE $08,$26,$02,$35,$B7,$A6,$8C,$D6
1700 'CODE $81,$08,$24,$0C,$AE,$8C,$D0,$A5
1710 'CODE $8C,$D1,$26,$04,$86,$20,$20,$02
1720 'CODE $86,$21,$A7,$8C,$13,$E6,$8C,$BA
1730 'CODE $4F,$E3,$8C,$B4,$1F,$02,$EC,$8C
1740 'CODE $B1,$40,$ED,$8C,$B4,$E6,$8C,$B1
1750 'CODE $20,$04,$A6,$80,$20,$01,$4F,$A7
1760 'CODE $A5,$5C,$2D,$F4,$31,$A8,$50,$6A
1770 'CODE $8C,$A0,$26,$E9,$A6,$8C,$91,$8B
1780 'CODE $40,$A7,$8C,$8C,$20,$AB
1790 'INIG;PRINT CHR$(12)
1800 'LINE[424,0,639,199,0,1,2]
1810 'SYMBOL[476,16,1,1,7,4,"High"]
1820 'SYMBOL[476,24,1,1,7,5,"score"]
1830 'SYMBOL[508,32,1,1,7,1,"0"]
1840 'SYMBOL[476,48,1,1,7,5,"Score"]
1850 'SYMBOL[508,56,1,1,7,1,"0"]
1860 'SYMBOL[476,70,1,1,7,5,"Floor"]
1870 'RETURN
```

リスト 2　Newゴルゴ13 マシン語ダンプ・リスト

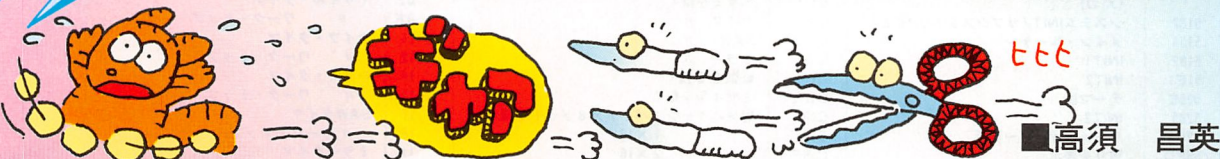
```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3C00 20 14 20 0C 20 47 01 DD 3C 00 3D 00 00 00 FF 00 :1D
3C10 34 1E 1A 50 20 0C 34 1E 1A 50 30 8C E3 CC 01 00 :10
3C20 30 8B 1F 50 1F 8B 9F 0A 30 8C D5 9F 08 0F 0C 0F :DF
3C30 0D 0F 0F 0D 0E 26 07 03 0E FC FF F8 DD 06 30 8C :16
3C40 48 BF FF F8 86 04 B7 FD 02 1C AF 35 9E 34 06 1A :30
3C50 50 8D 04 1C AF 35 86 CC 0D 00 8D 14 4A 2A FB CC :1C
3C60 07 FF 8D 0C 7F FD 02 0F 0E EC 8C 9A FD FF F8 39 :79
3C70 34 02 B7 FD 0E 86 03 B7 FD 0D 7F FD 0D F7 FD 0E :CD
3C80 4A B7 FD 0D 7F FD 0D 35 82 1F 50 1F 8B 9E 0A DE :EA
3C90 08 DC 0C 27 07 C3 FF FF DD 0C 20 43 EC 81 2B 32 :F5
3CA0 81 0E 26 06 36 14 DF 08 20 F2 81 0F 26 10 6D C4 :F5
3CB0 27 08 6A C4 26 04 33 43 20 EC AE 41 20 DE 81 0D :84
3CC0 26 02 D7 0F 8D AA 81 06 22 D2 86 0D D6 0F 8D A0 :65
3CD0 20 CA 81 FF 26 05 17 FF 7E 20 04 84 7F DD 0C 9F :D8
3CE0 0A 9E 06 6E 84 00 00 00 00 00 00 00 00 00 00 00 :A0
3CF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
----------------------------------------------------------
Sum: AE 2C A6 50 48 47 D3 1B ED E8 B1 46 CC 2E EE E8 :E9

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3D00 07 F8 0A 0D 05 04 04 16 80 42 0A 00 80 42 0A 0D :DE
3D10 80 21 0A 00 80 21 0A 0D 04 96 80 42 0A 00 80 42 :8B
3D20 0A 0D 80 21 0A 00 80 21 0A 0D 05 05 04 26 80 42 :70
3D30 0A 00 80 42 0A 0D 04 74 80 01 08 0E 00 82 80 21 :15
3D40 08 00 80 21 08 0E 80 85 0A 00 80 42 0A 0D 04 26 :D1
3D50 00 57 80 01 09 0C 02 6D 80 85 02 82 80 42 02 A4 :4D
3D60 80 42 0A 00 08 00 80 42 0A 0D 08 0E 04 74 00 61 :9C
3D70 80 01 02 74 80 42 02 92 80 85 08 00 02 AE 80 21 :AB
3D80 0A 00 80 21 0A 0D 08 0E 05 06 04 1F 02 82 00 6D :F7
3D90 80 64 0A 00 80 21 02 A4 80 42 02 C3 80 01 0A 0D :54
3DA0 80 20 09 00 80 01 0A 00 80 20 09 0C 80 01 0A 0D :81
3DB0 80 20 09 00 80 01 0A 00 80 20 09 0C 02 AE 80 01 :1A
3DC0 0A 0D 80 20 09 00 80 01 0A 00 80 20 09 0C 02 A4 :A6
3DD0 80 01 0A 0D 05 05 04 26 80 02 08 00 80 3F 09 00 :1E
3DE0 80 01 0A 00 80 85 09 0C 80 02 08 0E 00 74 80 C6 :F7
3DF0 0A 0D 05 04 04 96 02 C3 80 02 00 92 80 83 04 16 :B0
----------------------------------------------------------
Sum: 41 80 55 58 4E DE 43 26 31 8B D1 E1 2B CF 33 06 :A4

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3E00 02 AE 80 02 00 82 80 83 0A 00 80 42 0A 0D 80 21 :3B
3E10 0A 00 80 21 0A 0D 80 42 0A 00 80 42 0A 0D 80 21 :08
3E20 0A 00 80 21 0A 0D 80 03 09 00 80 02 08 00 80 3D :95
3E30 0A 00 80 42 0A 0D 80 21 0A 00 80 21 0A 0D 80 42 :08
3E40 0A 00 80 42 0A 0D 80 21 0A 00 80 21 05 00 04 00 :38
3E50 02 00 00 00 FF 07 F8 0A 0D 09 0A 08 0B 05 08 04 :4E
3E60 2C 02 DB 00 57 80 42 0A 00 09 00 08 00 80 42 0A :09
3E70 0D 09 0A 08 0B 80 21 0A 00 09 00 08 00 80 21 0A :9A
3E80 0D 09 0A 08 0B 80 42 0A 00 09 00 08 00 80 42 0A :DC
3E90 0D 09 0A 08 0B 80 21 0A 00 80 21 0A 0D 05 05 04 :A4
3EA0 74 80 21 0A 00 09 00 08 00 80 21 0A 0D 09 0A 08 :03
3EB0 0B 80 21 0A 00 09 00 08 00 80 21 0A 0D 09 0A 08 :9A
3EC0 0B 80 21 0A 00 09 00 08 00 80 21 0A 0D 09 0A 08 :9A
3ED0 0B 04 C7 02 E9 00 5C 80 21 0A 00 09 00 08 00 80 :59
3EE0 21 0A 0D 09 0A 08 0B 80 21 0A 00 09 00 08 00 80 :9A
3EF0 21 0A 0D 09 0A 08 0B 80 21 0A 00 09 00 08 00 80 :9A
----------------------------------------------------------
Sum: 56 63 BD 12 9C E8 B0 D4 A1 42 0E 2B 6A E4 D4 7F :4D

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3F00 21 05 00 04 00 02 00 00 00 FF 07 FE 08 10 0B 7F :D2
3F10 0C 0A 01 00 00 C8 80 03 00 C3 80 03 00 BE 80 03 :E9
3F20 00 B9 80 03 00 B4 80 03 00 AF 80 03 00 AA 80 03 :D2
3F30 00 A5 80 03 00 A0 80 03 00 9B 80 03 00 96 80 03 :82
3F40 00 91 80 03 00 8C 80 03 00 87 80 03 00 82 80 03 :32
3F50 00 7D 80 03 00 78 80 03 00 78 80 03 00 73 80 03 :EC
3F60 00 6E 80 03 00 69 80 03 00 64 80 03 08 00 FF 06 :D1
3F70 05 07 F7 08 10 0B 00 0C 0A 0D 00 80 7D FF 07 FE :4A
3F80 08 0F 00 18 80 0A 08 00 FF E5 7D FD 03 3B E5 E5 :27
3F90 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3FA0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3FB0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3FC0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3FD0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3FE0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
3FF0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
----------------------------------------------------------
Sum: 7D 42 BB 76 D3 E3 4B 61 4C A4 C7 D0 D3 80 B9 BA :9F

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
```

リスト 2　Newゴルゴ13 マシン語ダンプ・リスト

```
4050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
40A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
40B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
40C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
40D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
40E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
40F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-------------------------------------------------------
Sum: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4100 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
4110 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
4120 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
4130 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
4140 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
4150 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
4160 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
4170 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
4180 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
4190 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
41A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
41B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
41C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
41D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
41E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
41F0 FF FF FF FF FF FF FF FF FF FF FF F0 78 FF FF FF :5A
-------------------------------------------------------
Sum: F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 E1 69 F0 F0 F0 :6A

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4200 FF BF FF FF FF FF 1D FF FF FF E7 01 BF FF FF C7 :40
4210 06 1F FF FF C7 30 1F FF FF E7 41 0F FF FF 3F 80 :2B
4220 7F FF FE 10 0E BF FF FF 0A 0C 7F FF FF 84 44 3F :F1
4230 FF FF C1 0F FF FF FF F8 47 FF F9 FF FF F7 FF FA :F0
4240 00 00 00 00 00 00 00 00 00 00 00 0F 87 00 00 00 :96
4250 00 40 00 00 00 00 E2 00 00 00 18 FE 40 00 00 38 :B0
4260 F9 E0 00 00 38 CF E0 00 00 18 BE F0 00 00 C0 7F :C5
4270 80 00 01 EF F1 40 00 00 F5 F3 80 00 00 7B BB C0 :FF
4280 00 00 3E F0 00 00 00 07 B8 00 06 00 00 08 00 05 :00
4290 FF 80 00 3F FF FE 00 00 0F 07 F8 0F 87 FF 07 F0 :55
42A0 00 40 38 07 E0 00 E2 38 07 E0 18 FE 7F 07 E0 38 :14
42B0 F9 FF 07 F0 38 CF FF 07 F8 18 BE FF 07 FC C0 7F :0B
42C0 BF 07 FF EF F1 7F 07 FE F5 F3 FF 07 FF FB BF FF :CF
42D0 07 FC 3E F7 FF 07 FE 07 BF FF 01 FE 00 0F FF 00 :0E
42E0 FF FF FF FF FF FF FF FF FF FF FF FF 1E 0F FF FF :1F
42F0 FF FD FF FF FF FF B8 FF FF FF FD 80 E7 FF FF F8 :07
-------------------------------------------------------
Sum: B8 BA 76 16 01 4D 99 3E BC EB C6 9B 94 16 5F 99 :CD

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4300 60 E3 FF FF F8 0C E3 FF FF F0 82 E7 FF FF FE 01 :7C
4310 FC FF FF FD 70 08 7F FF FE 30 50 FF FF FC 22 21 :A8
4320 FF FF FF F0 83 FF 9F FF E2 1F FF 5F FF EF FF FF :58
4330 00 00 00 00 00 00 00 00 00 00 00 00 E1 F0 00 00 :D1
4340 00 02 00 00 00 00 47 00 00 00 02 7F 18 00 00 07 :E9
4350 9F 1C 00 00 07 F3 1C 00 00 0F 7D 18 00 00 01 FE :74
4360 03 00 00 02 8F F7 80 00 01 CF AF 00 00 03 DD DE :48
4370 00 00 00 0F 7C 00 60 00 1D E0 00 A0 00 10 00 00 :98
4380 FF FC 00 01 FF E0 F0 00 00 7F E0 FF E1 F0 1F E0 :F9
4390 1C 02 00 0F E0 1C 47 00 07 E0 FE 7F 18 07 E0 FF :D2
43A0 9F 1C 07 E0 FF F3 1C 0F E0 FF 7D 18 1F E0 FD FE :2D
43B0 03 3F E0 FE 8F F7 FF E0 FF CF AF 7F E0 FF FD DF :3C
43C0 FF E0 FF EF 7C 3F 80 FF FD E0 7F 00 FF F0 00 7F :D1
43D0 C3 00 00 C3 3C FF FF 3C C3 00 00 C3 00 81 81 C3 :47
43E0 FF 7E 7E 3C 00 81 81 C3 00 00 00 00 00 00 00 00 :FC
43F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-------------------------------------------------------
Sum: 7B B6 61 D9 22 A2 96 EA A3 0A 88 54 ED 34 77 02 :D2

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4400 00 00 00 55 55 54 2A AA AA 55 55 54 2A AA AA 55 :4D
4410 55 54 2A AA AA 00 00 00 00 00 00 00 00 00 00 00 :27
4420 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
4430 FF EA AB FF FF 55 55 7F FA AA AA BF F5 75 D5 57 :5E
4440 EA B6 B3 AF D5 0B A3 57 EB 05 41 2B D5 82 00 57 :E6
4450 E8 04 40 6F FE AA AA 9F FD 50 15 5F FF 88 0A FF :DD
4460 FF 44 15 FF FF E2 2B FF FF F5 47 FF FF EA AF FF :33
4470 00 00 00 00 00 00 00 00 00 00 00 00 00 20 84 00 :A4
4480 00 9C 99 00 00 4E E6 00 01 97 D3 80 00 BF 7B 00 :8E
4490 02 AE EA C0 03 FE FF C0 07 F8 1F E0 00 D8 1B 00 :0B
44A0 00 EC 37 00 00 36 6C 00 00 17 E8 00 00 1F F8 00 :DB
44B0 FF C0 03 FF FE 00 00 7F F8 00 00 1F E0 20 84 07 :E0
44C0 E0 9C 99 07 C0 4E E6 03 C1 97 D3 83 C0 BF 7B 03 :BE
44D0 E2 AE EA C7 FB FE FF DF FF F8 1F FF FF D8 1B FF :1E
44E0 FF FF EC 37 FF FF F6 6F FF FF F7 EF FF FF FF FF :64
44F0 FF 80 07 FF FF FC 00 FF FF F0 00 00 3F E0 00 80 1F :2E
-------------------------------------------------------
Sum: E6 E8 5B A6 87 04 9C 3D 3A 75 57 DB 70 9F E3 28 :2E

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +3 +C +D +E +F Sum
4500 C0 15 50 07 C0 0A 82 03 80 51 15 03 80 CD FE 83 :32
4510 80 70 75 03 C0 00 AA 03 E0 00 54 0F F8 00 A8 1F :D7
4520 FF 00 10 7F FF E0 A7 FF FF F0 4F FF FF F8 1F FF :65
4530 00 3B B8 00 01 DD DD 00 0E EE EE C0 1F 77 F7 E0 :C5
4540 2E BF F3 E8 36 8F C6 74 2E F3 3F B8 5D FD CF DC :E4
4550 3B FD FF B8 1D F9 FF 74 0E 70 FE E0 03 BF FD C0 :53
4560 00 5E BB 80 00 0F F0 00 00 07 E0 00 00 00 00 00 :7F
4570 FF AA AF FF FD 55 55 FF FA AA AA BF F5 55 D5 5F :88
4580 EA BF F2 AF D4 8F C6 57 AA F3 3F AB D5 CD CF D7 :99
4590 AB FD FF AB D5 F9 FF 57 EA 70 FE AF FA BF FD 5F :92
45A0 FF 1C 3A FF FF EB F7 FF FF F7 EF FF FF F8 1F FF :2D
45B0 FF FF FF FF FF FF FF FF FF FF FF FF FF F0 0F FF :F1
45C0 FF C7 E3 FF FF 3F FC FF FE 7F FE 7F FE 76 6E 7F :3C
45D0 FE 7E 7E 7F FE 7C 3E 7F FF 3F FC FF FF 9F F9 FF :7F
45E0 FF CF F3 FF FF E7 E7 FF FF F8 1F FF FF FF FF FF :9D
45F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-------------------------------------------------------
Sum: 36 6F 67 7D 73 C7 96 15 31 52 B1 FD B4 D5 BD 2D :12

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4600 00 00 00 00 00 00 00 00 00 00 00 00 00 36 6E 00 :A4
4610 00 00 06 00 00 00 06 00 00 00 0C 00 00 00 18 00 :30
4620 00 00 30 00 00 00 60 00 00 00 00 00 00 00 00 00 :90
4630 FF C0 03 FF FE 00 00 7F F8 00 00 1F E0 00 00 07 :3C
4640 C0 07 E0 03 C0 3F FC 03 E0 7F FE 07 E0 76 6E 07 :D7
4650 F0 7E 7E 0F F8 7C 3E 1F FC 3F FC 3F FF 1F F8 FF :57
4660 FF CF F3 FF FF E7 E7 FF FF F8 1F FF FF FF FF FF :9D
4670 FF E0 07 FF FE 00 00 7F F8 00 00 1F F8 00 00 1F :90
4680 00 00 00 00 00 00 00 00 C0 00 00 03 F8 98 19 3F :AB
4690 F8 61 06 3F F8 00 00 3F F8 02 80 3F F8 00 00 3F :C5
46A0 FC 00 00 7F FF 00 01 FF FF F0 0F FF FF FF FF FF :73
46B0 00 00 00 00 00 00 00 00 00 00 00 00 00 55 54 00 :A9
46C0 55 55 55 55 55 5F F5 55 16 03 C0 54 07 9F F9 C0 :DE
46D0 03 FC 7F 80 05 F8 3F 40 05 CC 67 40 07 FC 7F C0 :34
46E0 03 E0 0F 80 00 DE F6 00 00 0F F0 00 00 00 00 00 :45
46F0 FF E0 07 FF FE 00 00 7F F8 00 00 1F F8 00 00 1F :90
-------------------------------------------------------
Sum: FB 66 81 21 02 D7 B2 71 95 86 CB 77 AB 51 CF 47 :6E

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4700 00 00 00 00 00 0A A0 00 C2 01 40 43 FA 9A B9 BF :FC
4710 F9 70 17 3F F8 AA AA 3F FD 41 05 7F FA A8 2A BF :97
4720 FD 40 05 7F FF 8A A3 FF FF F5 5F FF FF FF FF FF :3A
4730 E7 C3 81 81 00 00 81 C3 18 3C 7E 7E FF FF 7E 3C :F8
4740 E7 C3 81 18 3C 99 C3 00 FE 00 01 FE 00 00 1F :78
4750 E0 01 FE 00 00 7F 80 0F FF F8 00 FC 00 01 FF 00 :E0
4760 00 00 00 54 01 54 2A A0 0A 54 01 54 2A 80 2A 50 :4A
4770 00 04 2A 02 AA 00 00 00 00 FE 00 01 FE 00 00 1F :F6
4780 E0 01 FE 00 00 7F 80 0F FF F8 00 FC 00 01 FF 00 :E0
4790 FE 07 00 01 C0 F1 E3 80 00 0F F0 E0 00 00 07 F8 :F8
47A0 00 00 00 03 C0 00 00 00 01 80 00 00 00 F0 C0 00 :F4
47B0 00 00 3C 00 00 00 00 02 00 00 00 00 00 00 00 00 :3E
47C0 00 00 00 00 00 00 01 00 00 00 00 03 00 00 00 00 :04
47D0 71 E0 00 20 03 F8 F8 00 30 30 7E FC 60 3E 1E 3F :39
47E0 01 F8 FF FE 3F 0E 1C 7F FF F0 0F 1F FF FF F8 07 :F8
47F0 FF FF FF FC 3F FF FF FF FE 7F FF FF FF 0F 3F FF :FC
-------------------------------------------------------
Sum: F3 1A 7E 34 BB C2 28 82 0C E1 9F 8A 76 FE A4 84 :98

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4800 FF FF C3 FF FF FF FF FD FF FF FF FF FF FF FF FF :B2
4810 FF FF FF FF FF FF FF FE FF FF FF FC FF FF FF FF :EC
4820 8E 1F FF DF FC 07 07 FF CF CF 81 03 9F C1 E1 C0 :B7
4830 FE 07 03 F1 C0 F1 E3 8F 1E 0F F0 E0 00 07 C7 F8 :DF
4840 00 60 00 E3 C1 8A 46 01 E1 83 15 83 80 F0 C6 00 :07
4850 18 C0 3C 0C 0E 3C 03 C2 0F 18 33 80 E0 3F 86 00 :AE
4860 60 70 0F 00 03 81 C1 01 E0 18 33 03 03 C0 00 1C :32
4870 71 E0 7E 20 03 F8 F8 1F F0 30 7E FC 67 FE 1E 3F :5D
4880 FF FF C7 FF FF FF FE 0F F8 3F FF FC 07 E0 FF E3 :CA
4890 FC 03 F0 7F E1 FF 00 F0 3F E0 70 00 00 FF F8 00 :C4
48A0 00 00 3F E0 00 00 10 07 E0 00 00 47 C7 FF 00 00 :23
48B0 01 FF FF E0 00 00 73 FF 00 00 40 43 F0 07 00 78 :43
48C0 03 C0 0F E0 0E 1F E0 7F F8 1F 1F FF E0 FF 3F FF :90
48D0 00 00 38 00 00 00 01 F0 07 C0 00 03 F8 1F 00 1C :26
48E0 03 FC 0F 80 1E 00 FF 0F C0 1F 8F FF FF 00 07 FF :2C
48F0 FF FF C0 1F FF FF EF F8 1F FF FF B8 38 00 FF FF :CD
-------------------------------------------------------
Sum: 74 50 98 9A 9A 51 39 E8 A0 DB C4 1F 53 97 0C C5 :1B

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
4900 FE 00 00 1F FF FF 8C 00 FF FF BF BC 0F F8 FF 87 :AD
4910 FC 3F F0 1F F1 E0 1F 80 07 E0 E0 00 00 1F 00 C0 :60
4920 FF FF C7 FF FF FF FE 0F F8 3F FF FC 27 E0 FF E3 :EA
4930 FC 23 F6 7F E1 FF 10 F2 3F E0 70 EC 00 FF F8 0C :F4
4940 03 CC 3F E0 7E 08 10 07 E0 3C 1C 47 C7 FF 06 0E :E4
4950 01 FF FF E1 8B 06 73 FF 01 D0 46 43 F1 87 0C 78 :39
4960 03 C7 0F E2 0E 1F E0 7F F8 1F 1F FF E0 FF 3F FF :99
4970 00 00 00 C0 01 40 08 10 00 20 02 60 01 00 00 00 :9C
4980 00 40 41 E0 37 E0 1F FF 7F F0 07 F8 0D 8E 10 00 :AF
4990 00 00 00 00 00 80 07 E0 07 C0 01 80 00 00 00 00 :AF
49A0 D4 7F FF D5 55 50 00 00 00 00 00 08 00 00 00 00 :D4
49B0 00 00 00 00 00 00 55 54 7F FF FF FF F9 27 FF FF :43
49C0 A7 FF FF AA AA A8 00 00 00 00 01 40 00 00 00 00 :E2
49D0 00 00 00 00 00 02 AA AB FF FF FF 3C FF FF FF FF :89
49E0 7F FF FD 55 55 40 00 00 00 00 2A 80 00 00 00 00 :0F
49F0 00 00 00 00 00 15 55 53 FF FF FF FF DF FF FF FF :95
-------------------------------------------------------
Sum: F6 B0 36 D3 73 F9 9E 44 19 F6 C1 CA 0F 0F 14 F8 :C1
```

フライング・シザース

FLYING SCISSORS

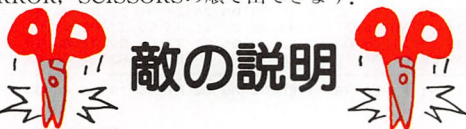
ナイフ、ハサミ、ガラスの破片がふってくる。あなたはもう先端恐怖症！

■高須　昌英

このゲームは，ナイフ，丸ノコ，ガラス，ハサミをよけながら，ひたすら進むゲームです．キャラクタによって動きのパターンが違うので，かなりの反射神経が必要です．

ゲームについて

4，6のキーで船を動かして，SPACEキーでビームを発射します．このビームは二連装で，4発まで画面に存在します．

敵は，3500点周期で，点数に応じてROUND，SAW，MIRROR，SCISSORSの順で出てきます．

敵の説明

●KNIFE（50点）

常に上から下へ落ちてきます．なかなか当たりません．

●ROUND SAW（70点）

ジグザグに降りてきます．

●MIRROR（150点）

6つ束になって，高速で前方から突っ込んできます．

●SCISSORS（100点）

絶対に避けられません．早目に撃ち落としましょう．

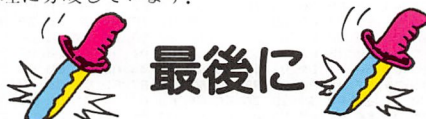
プログラムについて

プログラムは，$5000～60C4で，スタートは$5000ですが，△×00でリロケータブルです．

音出力は，PSGがなければスピーカから音が出るようになっています（519Bでフラグを作っているので，音がでないときは，そこを変えてください）．

それから，1.2MHzのモードで動かさないと，MIRRORが速すぎてゲームになりません．

サブシステムに送ったプログラムは，**JMP〔$D380〕**で各処理に分岐しています．

最後に

このプログラムはエディタ・アセンブラV2.0を使って作りました．ソース・プログラムが，MAIN側で12Kバイトを越えましたが，ALL RAMの力が発揮されました．FM-8の人もRAM FILEの改造をされたらどうでしょう．

しかし，これを作っていて，ディスクやプリンタが欲しくなりました．ただで誰かくれないかなぁ．

参考文献

1) "ALIEN FALL" I/O'80年11月号
2) "CRASH" ゲームの本 4
3) "シーソーゲーム" I/O'82年10月号

写真1 初期画面

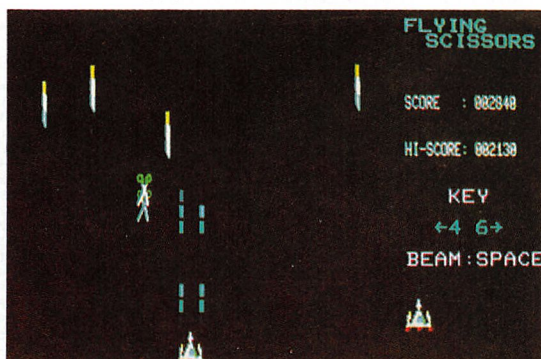
写真2 ナイフの間を丸ノコが…

FLYING SCISSORS

表1　エントリと処理

アドレス	動　作	アドレス	動　作
5000	サブシステムにSUB TOP以降を転送	589E	データ・エリア
5089	サブシステムHLT	5A75	サブシステムのプログラム
509F	一行出力(Uより)	C000	MAINとコマンドのやりとりをする
50B7	A＝乱数	C019	D401にキーコードを格納
50C7	スピーカーから音を出す(Uより)	C022	全画面消去
50D2	〃　(A, X)	C039	1行出力
50EC	数字表示　(D, X, <7)	C09F	VRAMアドレスを計算するサブルーチン
5107	PSGから音を出す(Uより)	C0C3	数字を表示する．
5113	〃　(A, B)	C12D	キーが押されるまでまつ
5126	サブシステムにコマンドと座標を送る	C136	キーを読んだあと4をクリア
	(X, D)	C145	ハサミを描く
5137	システムINIT.(サブシステムDPなど)	C151	ナイフ　〃
5154	メイン・ルーチン	C15C	SAW　〃
5192	INIT1	C168	ミラー　〃
51E1	INIT2	C173	戦闘機を描く
525E	テーマ曲	C17E	ミサイルを描く
5285	INIT3	C189	爆発パターン1バイト×8ドットを描く
52E6	ゲーム・オーバー表示	C194	1×16
53FD	残りを表示	C19F	2×16
530F	やられたときの表示・音	CIAA	4×16
535A	戦闘機	C225	1バイト×8ドットを消す
53A7	音出し	C22C	1×16
53E4	ミサイル	C233	2×16
554C	スコア表示	C23A	4×16
5555	爆発音	C28B	左側の画面を消す
5578	発射音	C2AE	データ・エリア
558D	ナイフ	C64F	
561F	ハサミ		
56DF	のこぎり		MAIN上では，5A75から
57CC	ミラー		

表2　ワーク・エリア

アドレス	動　作
DP:00	やられたかのフラグ
01	乱数ワーク
03	スコア
05	ハイ・スコア
07	桁数
08	SOUNDワーク
0A	戦闘機のこり
·0B	〃　タイマ
DC	〃　X
0D	〃　キーバッファ
0E	ミサイル・タイマ
0F	〃　ワーク・トップ
11	ナイフ・タイマ
12	〃　ワーク・トップ
14	ハサミ・タイマ
15	〃　ワーク・トップ
17	SAWタイマ
18	〃ワーク・トップ
1A	ミラータイマ
1B	〃　ワーク・トップ
1D	〃　発生カウンタ
1E	ACH用PSGデータ・アドレス
20	PSGがついているかのフラグ　YES→0
21	BCH用PSGデータ・アドレス
23	ACH用タイマ
34	BCH　〃
25	ワーク(汎用)
30	各物体のワーク・エリア
∫	

FLYING SCISSORマシン語ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5000 B6 FD 05 2B FB 86 80 B7 FD 05 B7 FD 05 CE FC 80 :A0
5010 30 8C 3A C6 3D A6 80 A7 C0 5A 26 F9 7F FD 05 16 :96
5020 01 15 34 06 B6 FD 05 2B FB E6 61 86 80 B7 FD 05 :34
5030 B7 FD 05 BF FC FE 3A F7 FC 80 10 8E FC BC A6 C0 :DB
5040 A7 A0 5A 26 F9 8D 42 6A E4 26 DE 35 86 00 00 3F :DB
5050 59 41 4D 41 55 43 48 49 93 D3 8F 90 B6 D4 08 B6 :1E
5060 D4 09 7F D3 80 B6 D4 0A B6 D3 80 27 F8 B7 D4 0A :00
5070 B7 D4 0A 81 01 27 0F FE D3 FE 30 8C 0C E6 80 E7 :31
5080 C0 4A 26 F9 20 DC 7E C0 00 7F FD 05 B6 FD 05 2B :C7
5090 FB 86 80 B7 FD 05 B7 FD 05 B6 FC 80 26 EB 39 8D :7C
50A0 E8 E6 C0 8E FC 82 A6 C0 A7 80 5A 26 F9 CC C0 39 :65
50B0 FD FC 80 7F FD 05 39 DC 01 D3 01 D3 01 DD 01 9B :31
50C0 02 C3 00 81 DD 01 39 A6 C0 27 06 AE C1 8D 03 20 :0F
50D0 F6 39 97 08 86 81 98 09 97 09 B7 FD 03 D6 08 4F :FA
50E0 C0 01 26 FB 30 1F 26 EC 7F FD 03 39 34 02 8D 99 :57
50F0 35 02 FD FC 85 96 07 B7 FC 84 BF FC 82 CC C0 C3 :15
---------------------------------------------------------
Sum: B6 0A 48 AE E7 73 BE E6 33 C8 3E E0 90 71 57 98 :BD
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5100 FD FC 80 7F FD 05 39 A6 C0 81 FF 27 18 E6 C0 8D :8B
5110 02 20 F4 8E FD 0D A7 01 86 03 A7 84 6F 84 E7 01 :E5
5120 4A A7 84 6F 84 39 34 02 17 FF 5E 35 02 BF FC 80 :BD
5130 FD FC 82 7F FD 05 39 1F 50 4A 4A 1F 8B 1A 50 33 :7F
5140 8D 09 32 8E C0 00 CC 1B 3C 17 FE D6 86 01 B7 FC :5E
5150 80 7F FD 05 8D 3C 17 00 88 17 01 02 17 01 26 17 :D8
5160 01 F8 17 04 28 17 04 B7 17 05 74 17 06 5E 17 02 :32
5170 73 17 02 33 0D 00 27 E7 17 01 94 0A 0A 27 05 17 :DD
5180 01 7B 20 D8 17 01 5F DC 03 10 93 05 25 02 DD 05 :7B
5190 20 C4 8D 36 CC 00 00 DD 05 DD 08 7F FD 0E B6 FD :77
51A0 0E 81 FF 26 04 97 20 20 02 0F 20 1F B8 C6 30 DD :6A
51B0 0F C3 00 08 DD 12 C3 00 12 DD 15 C3 00 03 DD 18 :4B
51C0 C3 00 0A DD 1B C6 05 D7 07 39 8E C0 22 17 FF 56 :83
51D0 C6 07 33 8D 06 C8 34 04 17 FE C4 35 04 5A 26 F6 :1B
51E0 39 8E C2 8B 17 FF 3F 33 8D 07 1D C6 06 34 04 17 :68
51F0 FE AD 35 04 5A 26 F6 8E C1 51 CC 0D 07 17 FF 26 :16
---------------------------------------------------------
Sum: C5 1B A2 FA 53 00 0B F6 27 69 60 26 CE 5F B4 ED :B4
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5200 8E C1 45 CC 0D 0A 17 FF 1D 8E C1 5C CC 0C 0D 17 :51
5210 FF 14 8E C1 68 CC 0D 10 17 FF 0B CC 00 00 DD 03 :80
5220 8E 45 08 17 FE C6 DC 05 8E 45 0B 17 FE BE C6 03 :11
5230 8E 35 15 34 04 30 89 06 00 17 FE 4D CC C1 73 FD :2E
5240 FC 80 BF FC 82 7F FD 05 35 04 5A 26 E6 86 04 97 :FA
5250 0A 8E C1 2D 17 FE CF 17 FE 2F 7F FD 05 39 0D 20 :95
5260 27 09 33 8D 07 6C 17 FE 5E 20 19 33 8D 07 0C 17 :F9
5270 FE 95 8E 82 00 30 1F 26 FC 30 8D 07 55 34 10 11 :82
5280 A3 E1 26 EB 39 8E C2 8B 17 FE 9B 33 8D 07 6B 17 :A2
5290 FE 75 0F 00 86 03 97 0B 97 0E 97 11 97 14 97 17 :53
52A0 97 1A 97 23 97 24 0F 1D 86 1A 97 0C 0F 0D 33 8D :71
52B0 07 99 DF 21 33 8D 07 66 DF 1E DE 0F 86 FF A7 41 :24
52C0 33 42 11 93 12 26 F5 86 18 A7 42 33 43 11 93 18 :FF
52D0 26 F7 A7 42 33 45 11 93 1B 26 F7 4C C6 06 A7 42 :5B
52E0 33 43 5A 26 F9 39 33 8D 07 61 17 FD B2 17 FD AF :D9
52F0 8E C1 2D 17 FE 30 17 FD 90 7F FD 05 39 96 0A C6 :85
---------------------------------------------------------
Sum: 2D 41 1B 51 DC FB 4A 16 2C 5D 48 C9 10 70 6D C4 :5C
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5300 06 3D CB 35 1F 98 C6 15 8E C2 3A 17 FE 18 39 8E :53
5310 C1 AA 96 0C C6 17 17 FE 0D 0D 20 27 1B C6 07 34 :7C
5320 04 86 64 34 02 8E 00 64 17 FD A7 35 02 80 14 26 :C2
5330 F2 35 04 5A 26 E9 20 21 33 8D 07 30 17 FD C8 86 :2E
5340 64 34 02 17 FD 71 1F 89 86 06 17 FD C6 8E 0F A0 :6A
5350 30 1F 26 FC 35 02 4A 26 E8 39 0A 0B 27 01 39 86 :35
5360 02 97 0B 8E C2 3A 96 0C C6 17 17 FD B9 8E C0 19 :E1
5370 17 FD B3 17 FD 13 B6 FC 82 7F FD 05 81 20 27 02 :6D
5380 97 0D 96 0D 81 34 27 0E 81 36 26 10 96 0C 81 35 :76
5390 27 0A 0C 0C 20 06 96 0C 27 02 0A 0C 8E C1 73 96 :A8
53A0 0C C6 17 17 FD 80 39 8D 03 8D 1D 39 0A 23 27 01 :7E
53B0 39 86 03 97 23 33 8D 06 65 34 40 DE 1E 11 A3 E1 :AC
53C0 27 05 17 FD 42 DF 1E 39 0A 24 27 01 39 86 02 97 :66
53D0 24 33 8D 06 76 34 40 DE 21 11 A3 E1 27 05 17 FD :A8
53E0 26 DF 21 39 0A 0E 26 08 86 01 97 0E 8D 03 8D 48 :36
53F0 39 8E C1 36 17 FD 2F 17 FC 8F B6 FC 82 7F FD 05 :58
---------------------------------------------------------
Sum: 17 91 F1 C0 98 F1 E8 32 58 EC E1 CC 14 A6 AC 3D :90
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5400 81 20 26 33 DE 0F 5F A6 41 2A 01 5C 33 42 11 93 :CD
5410 12 26 F4 C1 02 25 20 17 01 5E 96 0C DE 0F C6 02 :01
5420 6D 41 2A 0F A7 C4 34 02 86 18 A7 41 35 02 8B 03 :D3
5430 5A 27 04 33 42 20 E9 39 DE 0F EC C4 5D 2B 0B 8E :FA
5440 C2 25 17 FC E1 5A E7 41 2A 0A 8E 00 C8 30 1F 26 :5C
5450 FC 16 00 EE 10 9E 12 6D A4 26 29 EC 21 10 A3 C4 :A4
5460 27 06 5C 10 A3 C4 26 1C 86 03 A7 A4 EC 21 8E C1 :72
5470 94 17 FC B2 DC 03 C3 00 05 DD 03 17 00 CE 17 00 :DC
5480 D4 16 00 BE 31 23 10 9C 15 26 CC 6D A4 26 30 EC :02
5490 21 A1 C4 27 05 4C A1 C4 26 25 E1 41 27 05 5C E1 :39
54A0 41 26 1C EC 21 8E C1 9F 17 FC 7B 86 03 A7 A4 DC :BC
54B0 03 C3 00 0A DD 03 17 00 93 17 00 99 16 00 83 31 :D4
54C0 23 10 9C 18 26 C5 6D A4 26 30 EC 21 A1 C4 2E 2A :03
54D0 8B 03 A1 C4 2D 24 E1 41 27 05 5C E1 41 26 1B EC :3D
54E0 21 8E C1 AA 17 FC 3F 86 03 A7 A4 DC 03 C3 00 07 :E9
54F0 DD 03 17 00 57 17 00 5D 20 48 31 25 10 9C 1B 26 :6D
---------------------------------------------------------
Sum: B8 4A AC 43 2E D3 94 89 54 41 D0 E4 51 C8 EB EE :4A
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5500 C5 C6 06 34 04 6D A4 26 2A EC 21 10 A3 C4 27 06 :DB
5510 5A 10 A3 C4 26 1D EC 21 8E C1 89 17 FC 08 86 06 :A0
5520 A7 A4 DC 03 C3 00 0F DD 03 17 00 20 17 00 26 35 :85
5530 04 20 0F 35 04 31 23 5A 26 C9 EC C4 8E C1 7E 17 :9D
5540 FB E4 33 42 11 93 12 10 26 FE EF 39 DC 03 8E 45 :18
5550 08 17 FB 98 39 86 FF A7 41 0D 20 27 11 86 1E 8E :EF
5560 00 64 17 FB 6D 86 0F 8E 00 64 17 FB 65 39 34 40 :8E
5570 33 8D 04 A1 DF 1E 35 C0 0D 20 27 0A 86 32 8E 00 :FB
5580 64 17 FB 4E 20 06 86 0D 5F 17 FB 87 39 0A 11 26 :EF
5590 08 86 03 97 11 8D 03 8D 32 39 DC 03 C3 00 96 DE :D7
55A0 12 34 06 A6 42 81 18 26 13 17 FB 0B 81 E1 25 0C :B6
55B0 17 FB 04 C6 39 3D C6 FF ED 41 6F C4 35 06 83 00 :36
55C0 32 25 07 33 43 11 93 15 26 D7 39 DE 12 EC 41 C1 :A1
55D0 18 26 09 8E 00 64 30 1F 26 FC 20 3B 8E C2 2C 6D :EE
55E0 C4 26 29 17 FB 40 5C ED 41 C1 18 27 2A 91 0C 2D :E3
55F0 11 80 03 91 0C 2E 0B C1 16 27 04 C1 17 26 03 0C :79
---------------------------------------------------------
Sum: B4 43 21 60 7D AC A8 24 89 7F 99 CA AF D7 8A E2 :CA
```

FLYING SCISSORSマシン語ダンプ・リスト

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5600 00 39 EC 41 8E C1 51 17 FB 1C 20 0B 6A C4 26 07 :BA
5610 17 FB 13 86 18 A7 42 33 43 11 93 15 26 AF 39 0A :F3
5620 14 26 08 86 03 97 14 8D 03 8D 41 39 DC 03 10 83 :7F
5630 01 5E 25 05 83 01 5E 20 F5 10 83 00 FA 24 01 39 :6B
5640 DE 15 E6 42 C1 18 26 0F 17 FA 6C 81 FA 25 08 8D :DB
5650 07 C6 FF ED 41 6F C4 39 17 FA 5C C6 28 3D 80 14 :92
5660 9B 0C 2A 01 4F 81 38 2D 02 86 37 39 DE 15 A6 42 :DA
5670 81 18 26 09 8E 00 64 30 1F 26 FC 20 3F 8E C2 33 :0D
5680 6D C4 26 2B EC 41 17 FA 9D 8D 32 C1 18 27 2D 4C :95
5690 91 0C 2D 11 80 04 91 0C 2E 0B C1 17 27 04 C1 16 :0F
56A0 26 03 0C 00 39 EC 41 8E C1 45 17 FA 79 20 0D 6A :50
56B0 C4 26 09 EC 41 17 FA 6E C6 18 E7 42 39 C0 18 50 :07
56C0 90 0C 2A 01 40 34 02 E1 E0 2F 05 EC 41 5C 20 0C :E7
56D0 EC 41 91 0C 2F 03 4A 20 03 27 01 4C ED 41 39 0A :4E
56E0 17 26 08 86 02 97 17 8D 03 8D 52 39 DC 03 10 83 :95
56F0 01 5E 25 05 83 01 5E 20 F5 10 83 00 32 25 3D 10 :B7
---------------------------------------------------------
Sum: A9 81 B1 4B E5 1F 2F 4C B2 52 3E 7E D2 6F 19 A8 :67
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5700 83 00 96 24 37 DE 18 A6 42 81 18 26 28 17 F9 A7 :F0
5710 81 F0 25 21 17 F9 A0 C6 33 3D 8B 02 5F ED 41 6F :26
5720 C4 17 F9 93 5F 44 59 58 5A E7 43 17 F9 89 C6 14 :B2
5730 3D 8B 0A A7 44 33 45 11 93 1B 26 CB 39 DE 18 EC :00
5740 41 C1 18 26 09 8E 00 C8 30 1F 26 FC 20 40 8E C2 :C0
5750 3A 6D C4 26 2C 17 F9 CE 8D 3C EC 41 C1 18 27 2E :BF
5760 8B 03 91 0C 2D 11 80 06 91 0C 2E 0B C1 17 27 04 :C8
5770 C1 16 26 03 0C 00 39 EC 41 8E C1 5C 17 F9 A7 20 :F4
5780 0D 6A C4 26 09 EC 41 17 F9 9C 86 18 A7 42 33 45 :42
5790 11 93 1B 26 AA 39 A6 41 AB 43 AB 43 A7 41 81 34 :28
57A0 2D 06 C6 FF E7 43 20 10 81 01 2E 06 C6 01 E7 43 :F9
57B0 20 06 6A 44 26 0C 60 43 17 F8 FC C6 14 3D 8B 0A :60
57C0 A7 44 17 F8 F2 81 E1 25 02 6C 42 39 0A 1A 26 08 :AE
57D0 86 01 97 1A 8D 03 8D 6D 39 0D 1D 26 41 DE 1B 0F :94
57E0 25 C6 06 A6 42 81 19 27 02 0C 25 33 43 5A 26 F3 :B6
57F0 0D 25 26 50 DC 03 10 83 01 5E 25 05 83 01 5E 20 :A5
---------------------------------------------------------
Sum: 96 12 3A 71 BC 80 06 44 6B 70 11 6C AB E7 86 1A :63
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5800 F5 10 83 00 96 25 3D 10 83 00 FA 24 37 17 F8 A7 :1E
5810 81 F0 25 30 86 06 97 1D 30 8D 02 02 9F 21 DE 1B :80
5820 96 1D C6 03 3D C0 03 33 C5 17 F8 8B C6 14 3D 80 :A5
5830 0A 9B 0C 2C 01 4F 81 38 2F 02 86 38 C6 FF ED 41 :C8
5840 6F C4 0A 1D 39 DE 1B C6 06 34 04 EC 41 C1 19 26 :BD
5850 09 8E 00 64 30 1F 26 FC 20 3C 8E C2 25 6D C4 26 :94
5860 2A 17 F8 C2 5C ED 41 C1 19 27 2B 91 0C 2D 12 80 :0D
5870 03 91 0C 2E 0C C1 17 27 04 C1 18 26 04 0C 00 35 :21
5880 84 EC 41 8E C1 68 17 F8 9D 20 0B 6A C4 26 07 17 :B1
5890 F8 94 86 19 A7 42 35 04 33 43 5A 26 AC 39 0B 02 :35
58A0 3B 03 05 46 4C 59 49 4E 47 00 0D 02 3E 04 05 53 :B5
58B0 43 49 53 53 4F 52 53 00 15 01 3B 08 07 53 43 4F :6B
58C0 52 45 20 20 20 3A 20 20 20 20 20 20 30 00 15 01 :37
58D0 3B 0B 07 48 49 2D 53 43 4F 52 45 3A 20 20 20 20 :41
58E0 20 20 30 00 08 02 41 0E 03 4B 45 59 00 0A 02 3F :00
58F0 10 01 1D 34 20 36 1C 00 0F 02 3B 12 03 42 45 41 :FD
---------------------------------------------------------
Sum: 72 EF 1B AC BF D9 A9 FD 97 21 E1 AD E0 D4 C5 E0 :05
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5900 4D 3A 53 50 41 43 45 00 10 02 12 03 06 53 43 4F :05
5910 52 45 20 54 41 42 4C 45 00 13 02 12 08 04 4B 4E :EB
5920 49 46 45 2E 2E 2E 2E 2E 2E 2E 35 30 00 13 02 12 :A2
5930 0B 04 53 43 49 53 53 4F 52 53 2E 2E 2E 31 30 30 :A3
5940 00 13 02 12 0E 04 52 4F 55 4E 44 20 53 41 57 2E :FA
5950 2E 2E 37 30 00 13 02 12 11 04 4D 49 52 52 4F 52 :DA
5960 2E 2E 2E 2E 2E 31 35 30 00 11 02 1D 17 02 50 55 :6A
5970 53 48 20 41 4E 59 20 4B 45 59 00 07 3C 08 0A 09 :0A
5980 0A 00 AE 01 00 02 5D 03 01 FF 00 9B FF 01 01 00 :B7
5990 25 03 02 02 4B FF 01 00 00 E9 FF 00 DB 03 01 02 :40
59A0 B7 FF 00 C3 FF 00 AE 02 5D FF 00 E9 FF 01 01 00 :6E
59B0 25 03 02 02 4B FF 01 00 00 AE FF 00 DB 03 01 02 :05
59C0 87 FF 00 C3 FF 01 01 00 05 03 02 02 0B FF FF 07 :66
59D0 3F FF 53 01 9B 4A 01 CA 8E 00 F5 70 01 34 69 01 :D4
59E0 48 5D 01 70 53 01 9B 70 01 34 8E 00 F5 53 01 9B :1C
59F0 69 01 48 5D 01 70 7E 02 26 00 07 38 08 0D 09 0B :8E
---------------------------------------------------------
Sum: 24 E1 E0 1F 06 63 E3 DF 53 1E 94 2E F1 D3 36 6F :CB
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5A00 0A 10 0C 02 0D 00 0B 00 00 00 01 00 02 00 03 00 :46
5A10 04 C8 05 00 FF 00 2B FF 00 24 FF 00 00 FF 02 09 :27
5A20 FF 02 0A FF 02 0C FF 02 0D FF 02 0E FF 02 10 FF :45
5A30 02 12 FF 02 13 FF 02 15 FF 02 18 FF 02 1B FF 02 :74
5A40 1D FF 02 20 FF 02 24 FF 02 00 FF 0E 02 11 0C 02 :92
5A50 47 41 4D 45 20 4F 56 45 52 00 11 02 0F 0E 03 50 :F9
5A60 55 53 48 20 41 4E 59 20 4B 45 59 00 07 37 08 10 :57
5A70 0C 28 0D 00 FF 7F D3 80 10 CE D0 00 B6 D4 0A B6 :0A
5A80 D3 80 27 F8 B7 D4 0A B7 D4 0A 6E FF D3 80 B6 D4 :E6
5A90 01 B7 D3 82 7E C0 00 8E 00 00 1F 13 EF 89 40 00 :C3
5AA0 EF 89 80 00 EF 81 8C 3E 7F 25 F1 7E C0 00 B6 D3 :8E
5AB0 84 C6 08 3D 86 50 3D 1F 01 F6 D3 83 3A 10 8E D3 :B9
5AC0 86 A6 A0 27 3F C6 08 3D CE D8 00 33 CB C6 08 34 :E3
5AD0 04 E6 C0 B6 D3 82 81 01 27 13 A6 5F 30 01 8D 27 :5B
5AE0 34 02 B6 D3 85 8D 3D 35 02 8D 1C 30 1F B6 D3 85 :4B
5AF0 8D 32 30 88 50 35 04 5A 26 D5 30 89 FD 80 F6 D3 :54
---------------------------------------------------------
Sum: 66 ED 86 77 11 98 7A 69 2C AA 96 7B A4 5C CD 4F :DF
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5B00 82 3A 20 BD 7E C0 00 46 56 57 46 56 57 46 56 57 :B0
5B10 46 56 57 39 B6 D3 83 C6 08 3D 86 50 3D 1F 01 F6 :6C
5B20 D3 82 3A 39 46 24 02 E7 84 46 24 04 E7 89 40 00 :BD
5B30 46 24 04 E7 89 80 00 39 8D DA 33 8D 00 10 86 05 :59
5B40 B0 D3 84 C6 05 3D 33 C5 10 BE D3 85 6E C4 CC 27 :52
5B50 10 8D 17 CC 03 E8 8D 12 CC 00 64 8D 0D CC 00 0A :AA
5B60 8D 08 CC 00 01 8D 03 7E C0 00 34 06 4F 34 02 1F :0E
5B70 20 10 A3 61 25 06 A3 61 6C E4 20 F5 1F 02 35 02 :20
5B80 CE D9 80 48 48 48 33 C6 C6 07 A6 C0 A7 84 A7 89 :86
5B90 40 00 A7 89 80 00 30 88 50 5A 26 EE 30 89 FD D1 :ED
5BA0 35 86 0F 04 96 04 27 FC 16 FE E3 96 04 27 05 0F :57
5BB0 04 16 FE DA B7 D3 82 7E C0 00 31 8D 01 65 17 FF :76
5BC0 53 C6 10 16 00 88 31 8D 01 B9 17 FF 47 C6 10 20 :92
5BD0 59 31 8D 01 DE 17 FF 3C C6 10 16 00 95 31 8D 02 :89
5BE0 92 17 FF 30 C6 08 20 42 31 8D 02 9F 17 FF 25 C6 :68
5BF0 10 20 7F 31 8D 03 54 17 FF 1A C6 06 20 2C 31 8D :CA
---------------------------------------------------------
Sum: E3 51 0E 30 77 B8 9B CC 5A 25 83 B9 53 7F D3 81 :E9
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5C00 03 5B 17 FF 0F C6 08 20 21 31 8D 03 68 17 FF 04 :D5
5C10 C6 10 20 16 31 8D 03 8D 17 FE F9 C6 10 20 2F 31 :BE
5C20 8D 03 E2 17 FE EE C6 10 20 48 34 14 8D 15 35 14 :E6
5C30 34 14 30 89 40 00 8D 0B 35 14 30 89 80 00 8D 03 :EB
5C40 7E C0 00 A6 A0 A7 84 30 88 50 5A 26 F6 39 34 14 :AE
5C50 8D 15 35 14 34 14 30 89 40 00 8D 0B 35 14 30 89 :C6
5C60 80 00 8D 03 7E C0 00 EE A1 EF 84 30 88 50 5A 26 :D8
5C70 F6 39 34 14 8D 15 35 14 34 14 30 89 40 00 8D 0B :3B
5C80 35 14 30 89 80 00 8D 03 7E C0 00 EE A1 EF 84 EE :40
5C90 A1 EF 02 30 88 50 5A 26 F2 39 17 FE 77 C6 08 20 :BF
5CA0 15 17 FE 70 C6 10 20 0E 17 FE 69 C6 10 20 1B 17 :44
5CB0 FE 62 C6 10 20 2A 4F A7 89 80 00 A7 89 40 00 A7 :96
5CC0 84 30 88 50 5A 26 F0 7E C0 00 CE 00 00 EF 89 80 :00
5CD0 00 EF 89 40 00 EF 84 30 88 50 5A 26 F0 7E C0 00 :E1
5CE0 CE 00 00 EF 89 80 00 EF 89 80 02 EF 89 40 00 EF :67
5CF0 89 40 02 EF 84 EF 02 30 88 50 5A 26 E6 7E C0 00 :DB
---------------------------------------------------------
Sum: CF 6B 48 2D B2 DF 13 2E 93 75 89 E4 88 29 EB 55 :E7
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5D00 8E 00 00 C6 C8 34 10 86 39 6F 84 6F 89 40 00 6F :B9
5D10 89 80 00 30 01 4A 26 F1 35 10 30 88 50 5A 26 E5 :4D
5D20 7E C0 00 00 00 00 00 00 00 00 00 00 00 06 60 03 :A7
5D30 C0 03 C0 03 C0 05 A0 0A 50 14 50 28 50 50 28 A0 :39
5D40 28 C0 18 00 00 00 00 00 00 00 00 00 00 06 60 03 :69
5D50 C0 02 40 03 C0 05 A0 0A 50 14 50 28 50 50 28 A0 :B8
5D60 28 C0 18 7C 0E 84 31 86 31 7E 3E 06 30 06 60 03 :51
5D70 C0 02 40 03 C0 05 A0 0A 50 14 50 28 50 50 28 A0 :B8
5D80 28 C0 18 00 00 00 00 00 00 00 7C 7C 7C 7C 7C 7C :E8
5D90 3C 1C 0C 78 78 78 78 78 78 78 70 70 70 70 70 70 :4C
5DA0 30 10 00 68 68 68 68 68 68 68 7C 7C 7C 7C 7C 7C :00
5DB0 3C 1C 0C 00 1B 30 00 01 AA AB 00 07 55 55 40 1A :10
5DC0 AA AA B8 15 55 55 50 6A AB EA AE 35 5C 35 54 EA :CC
5DD0 B0 0A AB 35 50 0D 54 EA AC 3A AF 15 57 D5 58 3A :9D
5DE0 AA AA AC 05 55 55 60 06 AA AA A0 01 D5 57 00 00 :36
5DF0 13 30 00 00 00 00 00 00 2A A8 00 01 55 55 00 02 :C2
---------------------------------------------------------
Sum: 0C 5D AF AA 0C D8 2B 56 44 3A 47 30 93 0F 12 E5 :B5
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5E00 AA AA A0 15 55 55 50 0A AB EA A8 35 5C 35 54 2A :8E
5E10 B0 0A A8 35 50 0D 54 2A AC 3A AC 15 57 D5 58 0A :A7
5E20 AA AA A0 01 55 55 40 02 AA AA 80 00 55 54 00 00 :5E
5E30 00 00 00 00 1B 30 00 01 AA AB 00 07 55 55 40 1A :AC
5E40 AA AA B8 15 55 55 50 6A AB EA AE 35 5C 35 54 EA :CC
5E50 B0 0A AB 35 50 0D 54 EA AC 3A AF 15 57 D5 58 3A :9D
5E60 AA AA AC 05 55 55 60 06 AA AA A0 01 D5 57 00 00 :36
5E70 13 30 00 00 FF 8F 4E 4E 3C 3C 18 00 F3 83 42 42 :F7
5E80 24 24 10 00 FF 87 46 46 2C 2C 18 00 03 00 00 00 :DD
5E90 02 00 00 00 07 80 00 00 0F C0 00 00 02 00 00 00 :5A
5EA0 01 00 00 00 0A 80 00 00 0D 40 00 28 3F 80 A0 14 :73
5EB0 3F C0 50 20 FF C0 20 10 FF C0 10 2B FA A0 A0 55 :E7
5EC0 5F D5 50 00 00 00 00 00 00 00 00 00 03 00 00 00 :87
5ED0 02 00 00 00 07 80 00 00 0F C0 00 00 02 00 00 00 :5A
5EE0 01 00 00 0C 0A 80 C0 0C 0D 40 C0 28 38 00 A0 14 :84
5EF0 30 00 50 20 F0 00 20 10 F0 00 10 2B FA A0 A0 55 :7A
---------------------------------------------------------
Sum: 13 A5 F7 E6 1E 74 7C 51 3B 6F E1 42 4D 57 5A 86 :45
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5F00 5F D5 50 F0 07 80 3C F0 03 00 3C 00 03 00 00 00 :69
5F10 03 00 00 00 00 00 00 00 00 00 00 00 03 00 00 00 :06
5F20 03 00 00 0C 0F C0 C0 0C 0F C0 C0 3C 38 70 F0 3C :49
5F30 37 B0 F0 30 F7 BC 30 30 FF FC 30 3F FF FF F0 7F :F1
5F40 F0 3F F8 00 00 00 00 00 00 00 00 3C 3C 3C 3C 3C :53
5F50 3C 00 00 00 00 00 00 34 34 34 34 34 34 42 5D 75 :88
5F60 68 8A 7D 53 36 40 58 31 48 02 05 53 02 42 5D 75 :79
5F70 68 8A 7D 53 36 00 C1 01 23 03 02 1A 21 62 12 02 :93
5F80 1A A0 98 09 85 14 FF 65 BB 2F 3A 3E BF 76 1E 12 :1F
5F90 3A E0 F8 69 85 14 FF 65 BB 2F 02 1A 39 62 12 02 :2D
5FA0 1A A0 98 09 85 08 18 00 00 00 00 00 00 00 00 00 :00
5FB0 00 00 00 01 00 00 80 84 00 81 B0 80 31 40 22 51 :9A
5FC0 C0 38 B8 03 0C 08 38 29 10 1B 70 2D 34 73 D8 1C :8B
5FD0 60 13 40 37 60 5F E1 CF 61 DD F0 89 B3 57 A6 73 :33
5FE0 D2 7A F9 03 7C 08 38 39 50 31 F8 91 7C 02 38 14 :11
5FF0 80 03 00 23 00 47 81 CF 01 DD B0 80 33 53 26 73 :6A
---------------------------------------------------------
Sum: 78 C0 4B AE F0 22 AD E0 E8 DA 5B F7 8F C8 16 5E :AF
```

FLYING SCISSORSマシン語ダンプ・リスト

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
6000	D2	7A	F9	03	7C	00	00	88	00	01	51	44	80	3B	60	C2	:BF
6010	40	04	40	68	2D	18	20	15	00	08	C0	18	50	8B	38	00	:59
6020	76	52	00	03	10	1C	06	00	88	50	20	08	10	70	38	0C	:C1
6030	40	28	24	A0	60	20	D3	21	D8	04	00	38	20	06	F4	CC	:9A
6040	10	00	E3	60	00	00	01	FA	C0	03	80	8C	20	31	02	E6	:56
6050	0C	00	48	7C	17	12	B0	3F	40	07	D8	5C	81	83	BF	8F	:B5
6060	56	EA	3E	6F	80	4C	BE	B8	C0	F3	E4	CC	21	52	38	7E	:BB
6070	54	04	24	B5	0A	38	D3	F8	78	00	00	50	A0	01	60	E8	:EF
6080	10	00	08	71	00	00	00	98	00	01	51	4C	80	3B	60	C6	:A0
6090	48	04	40	7C	3D	1A	20	35	40	09	C0	58	D0	8B	B8	02	:2A
60A0	76	52	31	87	10	1C	1F	08	88	50	A6	0C	10	72	3A	0E	:27
60B0	54	2C	2E	A5	62	38	D3	F1	D8	04	00	F8	20	06	F4	EC	:8B
60C0	10	00	E3	60	00	00	00	00	00	00	00	00	00	00	00	00	:53
Sum:	C0	68	74	87	69	58	4D	6D	38	B8	24	48	E2	81	63	37	:F7

RANDOM BOX

FM-7 アプリケーション・ソフト

■大森　智

FM-7用の簡単なプログラムが，いくつかできたので発表します．

自然対数の底を求める（リスト1）

10行のLに桁数を代入後，Kコンパイラでコンパイル後，実行してください．ただし，下の桁はあまり信用できません．また，あまり効率が良くないので，時間がかかります．10,000桁で約5時間．

リスト1　自然対数の底を求めるプログラム　ソース・リスト

```
10 ' CONST L=4050,K=10
20 ' A=$8000;P=$1300
25 ' POKE $FDOF,0
30 ' FOR I=0 TO L;
40 '   P:I)=0;A:I)=0;
50 ' NEXT
55 '
60 ' A:1)=1;P:1)=2;N=2
70 ' LAB2;IF N<3000 THEN GOTO LAB10;
75 '                 ELSE GOTO LAB11;FI
80 ' LAB120;GOTO LAB20
90 ' LAB130;N=N+1
100 '    IF A:L)=0 THEN GOTO LAB90;FI
110 '  GOTO LAB2;
115 '
120 'LAB90; I=L;REPEAT;
130 '                IF A:I)=0 THEN GOTO LAB91;FI;
140 '               GOTO LAB2;
150 'LAB91;          I=I-1;
155 '              UNTIL I=0
160 '  GOTO LAB30;
165 '
170 'LAB10; R=0;I=0
180 'REPEAT; I=I+1
190 '  X=A:I)+R*K;
200 '  A:I)=X/N;
210 '  R=X-N*A:I);
220 'UNTIL I=L
230 'GOTO LAB120
231 '
232 'LAB11; R=0;I=0
233 'REPEAT; I=I+1
234 '  Y1=R*5;Y2=Y1/N
235 '  X=Y1-Y2*N;X=X*2+A:I)
236 '  A:I)=X/N
237 '  R=X-N*A:I)
238 '  A:I)=A:I)+Y2*2
239 'UNTIL I=L
240 'GOTO LAB120
241 '
249 'LAB20; C=0;I=L
250 '   REPEAT
260 '       P:I)=P:I)+A:I)+C;
270 '       IF P:I)-K>=0 THEN GOTO LAB6;FI;
280 '       C=0
290 '       GOTO LAB4;
300 'LAB6;  C=1;
310 '       P:I)=P:I)-K;
320 'LAB4;  I=I-1;
325 '   UNTIL I=0
330 'GOTO LAB130;
335 '
340 'LAB30; PRINTCHR$(12),"E=",P:1),".";
350 '  FOR I=2 TO L
390 '    PRINT P:I);
400 '  NEXT
410 'END
```

Vol.Copy（リスト2）

シングル・フロッピーしかなかった人には，コピーをするとき，出し入れすること40回でした．その回数を14回にしました．

Kコンパイラでコンパイル後，実行する前には必ず，NEW↵CLEAR，&HFFF↵を実行してください。

なお，このプログラムは両面倍密40トラック，16セクタのものであれば，他社のディスクでもコピーできます．また，デュアル・ドライブにも対応し，DISK BASICなしでも走ります．

リスト2　vol.Copyプログラム　ソース・リスト

```
1000 'CODE $8E,$F000#,$F6,$FDOF#,$E684#,$F7,$FDOF#
1005 'CODE $E780#,$8C,$FC00#,$26F1#,$F6,$FDOF#
1010 'PRINT/"*****  NEW VOL.COPY *****"
1020 'PRINT/" 0.  DRIVE 0 => 0"
1030 'PRINT/" 1.  DRIVE 0 => 1"
1040 'PRINT/"          ( 0 / 1 )"
1050 'INPDR;DR=INPUT;IF DR<0 OR DR>1 THEN
1060 '                     PRINT CHR$(7)
1070 '                     GOTO INPDR
1080 '                FI
1085 '      PRINT//
1090 'IFDR=1THENPRINT"SET DSK FOR LOAD & SAVE OK"/
1100 '      OK[]
1110 'FI
1120 'FOR TR=0 TO 39 STEP6
1130 '     ST=TR;ET=TR+5
1140 '     IF TR=36 THEN ET=39;FI
1150 '     IF DR=0 THEN PRINT"SET DSK FOR LOAD OK"/
1160 '         OK[]
1170 '     FI
1180 '     DRI=0;MDE=10;TRCK[]
1190 '     IF DR=0 THEN PRINT"SET DSK FOR SAVE OK "/
1200 '         OK[]
1210 '     FI
1220 '     DRI=DR;MDE=9;TRCK[]
1230 '     PRINT"TRACK",TR,"-",ET," / COPIED"/
1240 'NEXT
1250 'RCBSET[8,0,0,0,0,0]
1260 'END
1270 '
1280 'TRCK ;FOR I=ST TO ET
1290 '          FOR J=1 TO 32
1300 '                K=(I-ST)*32+J-1
1310 '                L=(J-1)/16;M=J-L*16
1320 '                IF K>=58 THEN K=K+16;FI
1330 '                IF K>=96 THEN K=K+16;FI
1340 '                MEM=$1000+K*256
1350 '                RCBSET[MDE,MEM,I,M,L,DRI]
1360 '
1370 '          NEXT
1380 '      NEXT
1390 '      RETURN
1400 'OK;
1401 '       OK1   ;Z=GET
1410 '              IF Z<>'Y'
1420 '                   GOTO OK1
1421 '              FI
1430 '       RETURN
1440 'RCBSET;ADDR=$5700
1450 '  POKE $FDOF ,0
1460 '   CPOKE ADDR,%1,0,%2#,%3,%4,%5,%6
1470 '   CODE  $8E,$5700#,$BD,$F17D#
1480 '  Z=PEEK($FDOF)
1490 ' IF PEEK($5701)<>0 THEN PRINT"ERROR";END;FI
1500 'RETURN
```

FM-7マルチステートメント化（リスト3）

プログラムを組んでいて，『マルチステートメントにしておけばよかった』というときに，このプログラムをマージします．そしてRUN 60000↵とすると，マルチステートメント化する行番号を聞いてきます．

プログラムの実行が終わってからリストを見ると，行がつながっています．あまり長い行をマルチステートメント化すると，エディットできなくなり，プロテクトとして使えるでしょう．

リスト3　マルチステートメント化プログラム

```
60000 A=PEEK(&H33)*256+PEEK(&H34):INPUT"START=",C:INPUT"END=",D
60010 S=PEEK(A+2)*256+PEEK(A+3)
60015 IFC=S THEN E=A:GOTO 60100
60020 IF(C<S)OR(S=60000!)THEN PRINT"ERROR":STOP
60030 A=PEEK(A)*256+PEEK(A+1):GOTO 60010
60100 S=PEEK(A+2)*256+PEEK(A+3)
60110 IF(D<S)OR(S=60000!) THEN PRINT"ERROR":STOP
60115 IF D=S THEN F=A:GOTO 60200
60120 A=PEEK(A)*256+PEEK(A+1):GOTO 60100
60200 A=PEEK(E)*256+PEEK(E+1):POKE E,PEEK(F):POKE E+1,PEEK(F+1)
60210 IF A>F THEN END
60220 B=A:A=PEEK(A)*256+PEEK(A+1)
60230 FOR I=-1TO2:POKE(B+I),32:NEXT:POKE(B+3),58:GOTO60210
```

◘参考文献

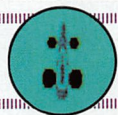
1）〝FM-7/8活用研究〞，工学社
2）大西正和：〝実用BASIC〞，日刊工業新聞

FM-7

インディ INDY-7

爆走インデアナポリス

■高畑英樹

FM-8用に〝FMラリー〟を作りましたが，これを発展させたのが〝INDY-7〟です．名前に深い意味はありませんが，きっとこの効果音のすごさに驚かれ，なるほどと思っていただければ幸いです．

プログラムについて

FM-7といえどもやはり多量のグラフィックを高速で移動させるには，サブシステムをユーザーが乗っ取り，直接画面制御をするプログラムをサブシステムで走らせる必要があるようです．そしてメインへの効果音出力依頼については，サブ側で高速割り込みを発生させて音の処理をしています．

また，BASICの**PLAY**文と**SOUND**文が使えるように，メイン側ではサブへのプログラム転送終了後，割り込み要求の受け入れを可能にし，各種の効果音をBASICレベルで実行しています．ですから音について気に入らない方は自由に変更が可能です．

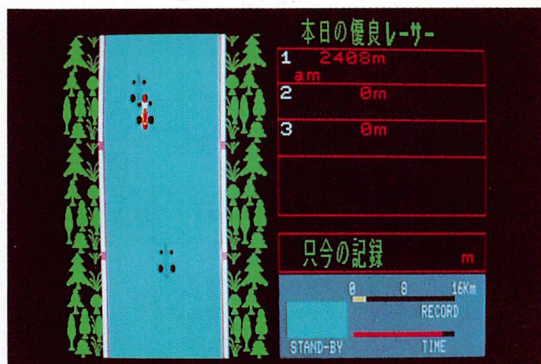
ゲーム中『本日の優良レーサー』の処理のため，プレイヤーの名前の入力，さらには記録などができるようにしました．そのため，**INPUT**文，**PRINT**文についてもゲームを中断することなくBASICレベルで実行し，再びサブのプログラムへ戻ってゲームが続行できるようにしています．

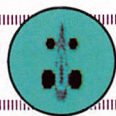
さらに，ゲーム中いかんを問わず，常に BREAK キーが押されると完全にBASICに復帰し，表示を含むすべてのBASICコマンドが実行できます．

これは，サブシステムでRUNさせるプログラム開発中においての有効な手段となり，画面状態は，そのままで作業ができ，能率が向上します．

SUBシステムのDPR

DPR（ダイレクト・ページ・レジスタ）は高速処理には欠くことのできないレジスタですが，システムで決定したアドレスから分離してもなおユーザー側で自由に使えないエリアがあります．メモリ・マップ（**表1**）を参考にしてください．

BREAK キーの入力情報がDPRの0番地にきているので，このエリアを常に参照していればプログラムを中断させることができます．

迫熱のデッドヒート！

キー入力情報エリアはD360ですが，D401と比べウェイト可能でキー入力をソフトでキャンセルできます．

ゲームの内容

ゲームセンターなどで古くからおなじみのカーレース・ゲームをマイコンで実現してみようと思い着手しましたが，スケール的に本物にはおよびませんでした．スピード感だけは表現できたと思います．

さて，FM側の車の動きはプレイヤー側の車の存在を無視し，常にランダムに方向転換などをします．しかしFM側の車同士では衝突しないようにしています．

ゲームをしていて，しばしば相手が寄ってきますが，決して故意ではないので安心して振り切ってください．目的は一定時間（4分）内でどれだけ多く走れるかを競うゲームです．もちろん時間内でも別の車に当ったときと，ガードレールに当ったときはプレイヤーの車が1つ失なわれ，計3回でゲーム・オーバーになります．

記録の結果は多数のプレイヤーと比較して上位3名についてのみ氏名を入力し『本日の優良レーサー』のコーナーに名を残すことになります．

当然ですが， BREAK キーでゲームを中断すると変数エリアが空白になり，再ゲームでは前回のゲームの名前などは消えてしまいます．

なお，プレイ・キーについては，TV画面で説明してい

るのでここでは省略します．BASICからRUNすると，デモンストレーションに入りますから，適当なキー入力でゲームがスタートします．

表1　サブシステムのメモリ・マップ（FM-7）

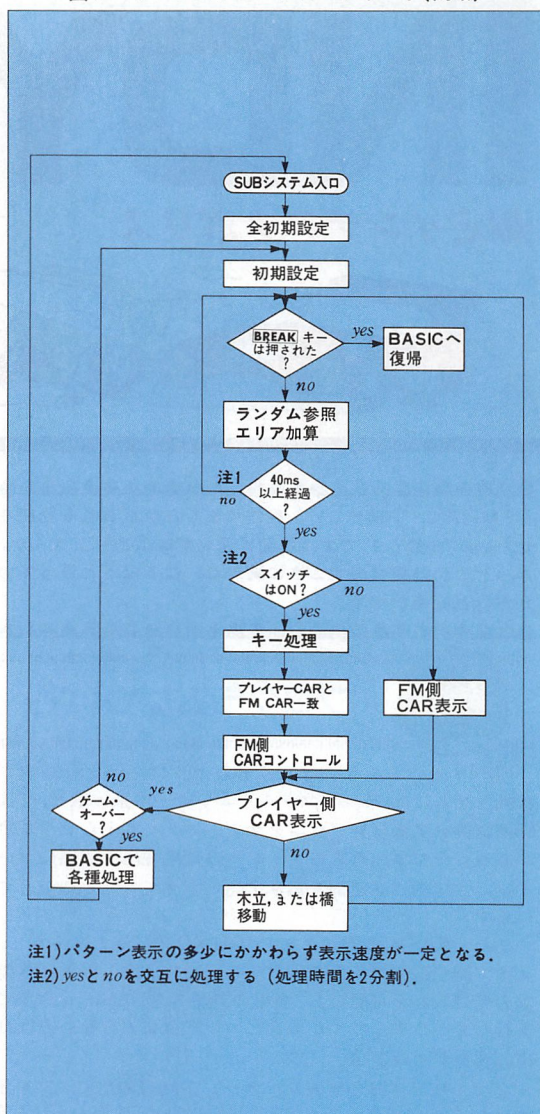
（ ）内はユーザー設定のDPR

注1：システムの決定したアドレスから分離したDPR上ではこの情報は得られない（D00C～D00Fを参照）．

注2：28H，29Hは使えない？

図1　サブシステム・フローチャート（概略）

注1）パターン表示の多少にかかわらず表示速度が一定となる．

注2）*yes*と*no*を交互に処理する（処理時間を2分割）．

INDY-7　BASICプログラム・リスト

```
10 COLOR7,0:WIDTH40,20:LOCATE13,3:PRINT"***PLAY
KEY***":LOCATE4,7:PRINT"5 カソク   2 ゲンソク   1 ヒダリ
  3 ミギ":LOCATE4,9:PRINT"ソレゾレ ノ ホーコー ノ テイシ ハ ス
ペースキー":LOCATE4,11:PRINT"キーボード ミギ ノ テンキー デ フ
°レイ シテクダサイ":LOCATE20,17:PRINT"HIT any key":I$=I
NPUT$(1)
20 CLS:DIMA%(15):POKE&H6FFF,0:LINE(350,153)-(608
,199),PSET,1,BF:SYMBOL(363,190),"STAND-BY",1,1.7
:SYMBOL(525,170),"RECORD",1,1,7:SYMBOL(525,190),
"TIME",1,1,7:SYMBOL(435,157),"0        8        16
Km",1,1,7:GOSUB230
30 SYMBOL(480,12),"レーサー",2,2,4:LINE(350,28)-(608
,120),PSET,2,B:LINE(350,130)-(608,151),PSET,2,B:
J=1:FORI=3TO8STEP2:LOCATE22,I:PRINTUSING"#";J:J=
J+1:NEXT:A=0:B=0:C=0:A$="":B$="":C$=""
40 LINE(440,165)-(567,167),PSET,0,BF
50 LINE(440,185)-(567,187),PSET,2,BF:POKE&H5C02,
0
60 EXEC&H5000
70 I=PEEK(&H6FFF):IFI<>0THENPOKE&H6FFF,0ELSE70
80 IFI=(&H80)THENSYMBOL(150,100),"START",3,2,3:L
INE(495,140)-(575,150),PSET,0,BF:POKE&H5C02,&H80
:FORI=0TO2000:NEXT:GOTO60
90 IFI=(&H81)THENSOUND8,0:SOUND9,0:SOUND10,0:SYM
BOL(136,100),"GAME OVER",2,2,2:I=PEEK(&H6FFD)*10
00:J=PEEK(&H6FFE):J=J*4:I=I+J:LOCATE31,14:COLOR2
:PRINTUSING"#####m";I:COLOR7:FORJ=0TO7000:NEXTEL
SE180
100 IFI>A THENC=B:B=A:A=I:C$=B$:B$=A$:GOSUB170:A
$=I$:GOTO140
110 IFI>B THENC=B:B=I:C$=B$:GOSUB170:B$=I$:GOTO1
40
120 IFI>C THENC=I:GOSUB170:C$=I$:GOTO140
130 GOTO40
140 COLOR2:LOCATE24,3:PRINTUSING"#####m";A:LOCAT
E23,4:PRINTUSING"&                &";A$
150 LOCATE24,5:PRINTUSING"#####m";B:LOCATE23,6:P
RINTUSING"&                &";B$
160 LOCATE24,7:PRINTUSING"#####m";C:LOCATE23,8:P
RINTUSING"&                &";C$:FORY=48TO88STEP20:
LINE(350,Y)-(608,Y),PSET,2:NEXT:LINE(351,90)-(60
7,119),PSET,0,BF:COLOR7:GOTO40
170 LOCATE22,10:PRINT"IN PUT your name":LOCATE22
,11:I$=INKEY$:INPUTI$:RETURN
180 IFI<11THENCAR=I:I=11-I:SOUND0,I*3:SOUND1,1:S
OUND6,&H1F:SOUND7,0:SOUND8,&H10:SOUND11,I*15+15:
SOUND12,0:SOUND13,8
190 IFI=12THENSOUND0,20:SOUND1,0:SOUND6,7:SOUND7
,0:SOUND8,15:FORI=0TO200:NEXT:SOUND8,0:I=CAR:GOT
O180
```

INDY-7 BASICプログラム・リスト

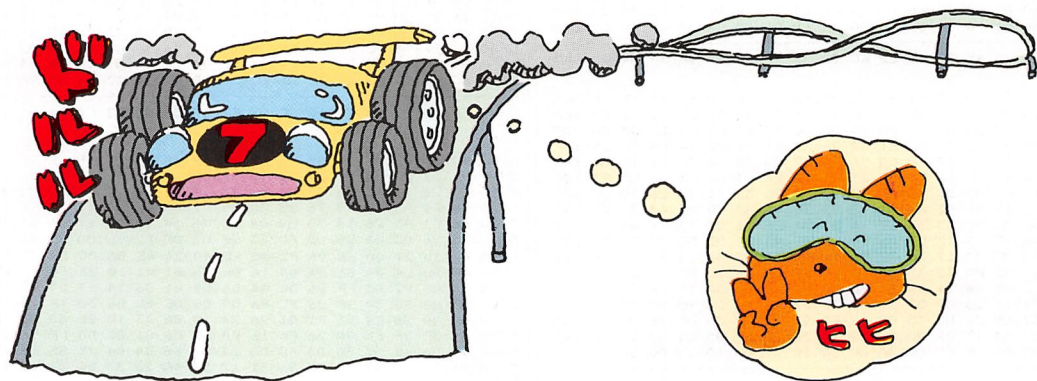
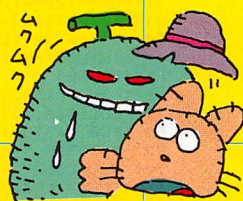
```
200 IFI=13THENCAR=15:SOUND7,7:SOUND6,&H1F:SOUND8
,&H1Ø:SOUND11,1:SOUND12,6Ø:SOUND13,Ø:GOTO7Ø
210 IFI=14THENPLAY"V15T1ØØL32":FORI=1TO4:PLAY"O5
CDE":FORJ=ØTO5Ø:K=PEEK(&H6FFF):IFK=13THEN7ØELSEN
EXTJ:NEXTI:I=CAR:J=PEEK(&H5CØ2):IFJ=ØTHEN7ØELSE1
8Ø
22Ø GOTO7Ø
23Ø RESTORE24Ø:FORX=38ØTO479STEP2Ø:FORI=ØTO15:RE
ADA%(I):NEXTI:PUT@(X,1Ø)-(X+15,25),A%,PSET,4:NEX
TX:FORX=38ØTO479STEP2Ø:FORI=ØTO15:READA%(I):NEXT
I:PUT@(X,133)-(X+15,148),A%,PSET,4:NEXTX:RETURN
24Ø DATA256,256,256,-2,256,768,896,1344,2336,436
8,2484Ø,-325Ø6,8176,256,256,256
25Ø DATAØ,Ø,8184,41Ø4,41Ø4,41Ø4,41Ø4,8184,41Ø4,4
1Ø4,41Ø4,41Ø4,41Ø4,8184,41Ø4,Ø
26Ø DATAØ,Ø,2Ø16,2Ø64,436Ø,8456,87Ø8,169ØØ,169ØØ
,17412,17416,1844Ø,12336,448,Ø,Ø
27Ø DATA4Ø96,6142,832Ø,1638Ø,25Ø92,25596,-24Ø6Ø,
-2Ø482,1Ø41Ø,8832,9468,8448,92Ø8,11536,8416,12Ø6
2
28Ø DATA256,256,8184,41Ø4,41Ø4,8184,41Ø4,41Ø4,81
84,4Ø96,4616,4368,4256,5696,6192,-8178
29Ø DATAØ,16376,82ØØ,82ØØ,82ØØ,82ØØ,82ØØ,82ØØ,16
376,Ø,Ø,2Ø8Ø,4112,82ØØ,-16378,Ø
3ØØ DATA256,64Ø,1Ø88,2Ø8Ø,4112,82ØØ,-12314,Ø,Ø,3
276Ø,8,16,16,32,64,896
31Ø DATAØ,Ø,2Ø16,2Ø64,436Ø,8456,87Ø8,169ØØ,169ØØ
,17412,17416,1844Ø,12336,448,Ø,Ø
32Ø DATA28672,5Ø8,4,-1Ø2Ø,4,4,-2Ø44,5Ø8,-1788,25
6,256,-1792,-3Ø464,-3Ø462,-179Ø,-3Ø466
33Ø DATA4Ø96,1Ø492,17412,-32764,31996,41ØØ,41ØØ,
-514,4112,4244,21592,2152Ø,21592,4244,7442,-8144
```

INDY-7 マシン語プログラム・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5000 B6 FD 05 2B FB 1A 50 B6 80 B7 FD 05 B7 FD 05 CE :8E
5010 FC 80 30 8D 00 84 C6 4A A6 80 A7 C0 5A 26 F9 7F :52
5020 FD 05 7F FD 05 8E 00 C8 30 1F 26 FC CE 5A 9D 8E :9D
5030 D1 00 86 09 C6 31 8D 21 7F FD 05 7F FD 05 33 8D :C7
5040 00 F8 8E BE 80 86 54 C6 34 8D 0E 86 01 B7 FC 80 :ED
5050 7F FD 05 7F FD 05 16 00 89 DD 00 B6 FD 05 2B FB :5C
5060 D7 01 86 80 B7 FD 05 B7 FD 05 BF FC FE 3A 10 8E :E1
5070 FC CA A6 C0 A7 A0 0A 01 26 F8 F7 FC 80 7F FD 05 :90
5080 7F FD 05 B6 FD 05 2B FB 86 80 B7 FD 05 B7 FD 05 :D7
5090 B6 FC 80 26 E8 0A 00 26 C7 39 00 00 3F 59 41 4D :96
50A0 41 55 43 48 49 93 D3 8F 90 B7 D4 0A B6 D4 08 B6 :CC
50B0 D4 09 B6 D4 0A 7F D3 80 7F D3 80 7F D3 80 B6 D4 :71
50C0 0A B6 D3 80 27 F8 81 01 27 15 B7 D4 0A B7 D4 0A :1A
50D0 FE D3 FE 8E D3 CA E6 80 E7 C0 4A 26 F9 20 D6 7E :E4
50E0 BE 8F 30 8D 00 06 BF FF F6 1C 00 39 34 76 B6 FD :76
50F0 04 84 02 26 1A 8E 01 E0 BF FF F6 86 40 B7 FD 05 :6C
-----------------------------------------------------------
Sum: E6 35 7A F4 ED FC 14 C7 D4 ED 95 B3 9C 5F 5B DC :88

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5100 B7 FD 05 B7 FD 05 8E 8F E1 AF 69 35 76 3B 39 86 :2D
5110 80 B7 FD 05 B7 FD 05 B7 FD 05 BE FC FD B6 FC FF :13
5120 7F FD 05 7F FD 05 4D 27 E2 BF 6F FD B7 6F FF 4D :F5
5130 2A D9 CC 01 E0 FD FF F6 20 D1 86 D0 1F 8B 8E C0 :E1
5140 00 6F 80 8C CF A0 26 F9 39 B7 D4 0A B7 D4 0A 86 :F2
5150 7F 1F 8B 16 00 9C 86 B4 C6 0A EF 81 5A 26 FB 30 :00
5160 88 3C 4A 26 F3 39 8E 06 50 CE FF FF 8D E8 8E 86 :99
5170 50 8D E3 8E 46 50 CE 00 00 8D DB 8E 06 4A 86 B4 :32
5180 C6 1A EF 84 EF 02 EF 04 3A EF 84 EF 02 EF 04 30 :F8
5190 88 36 4A 26 ED 8E 86 4A 10 8E 46 50 86 B4 97 19 :97
51A0 17 07 64 39 0F 01 0F 02 0F 03 4F C6 06 DD 1E CC :D0
51B0 D1 00 DD 20 DD 22 0F 25 86 01 C6 50 DD 26 86 01 :28
51C0 97 28 0F 31 0F 32 86 87 C6 1A DD 33 CC 00 00 DD :E6
51D0 35 DD 39 DD 3D 86 03 97 FD 97 FE 86 14 97 FA 97 :D9
51E0 FB 86 03 97 F7 86 01 97 F8 97 F5 17 FF 78 7F D3 :94
51F0 60 39 B6 D2 65 97 F3 86 34 97 F1 86 64 97 46 86 :9F
-----------------------------------------------------------
Sum: 94 FC 86 0C 09 51 F7 C6 FD C0 59 C1 9B 63 D9 65 :4C

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5200 01 97 49 86 FF 97 4E CC 7A 16 DD 47 CC 73 C7 DD :AE
5210 4A 0F 4C 0F 4D 8D 8D 96 F3 27 1C 86 A8 D6 F1 34 :10
5220 04 C0 03 D7 F1 35 04 17 02 2C 86 01 B7 D3 FF B6 :D3
5230 D4 04 B6 D4 04 20 17 86 A8 C6 2E DD 33 17 01 51 :38
5240 C6 31 D7 34 17 01 4A C6 34 D7 34 17 01 43 96 00 :5A
5250 27 03 7E BE 80 0C 30 B6 D0 0F 81 02 25 F0 7F D0 :9E
5260 0F 0C FF 96 FF 84 01 26 5B 96 F3 10 27 01 57 0A :D7
5270 46 26 1D 86 C8 97 46 9E 47 6F 84 6F 88 50 6F 89 :CB
5280 00 A0 30 1F 8C 7A 06 10 27 05 1F 9F 47 16 01 36 :89
5290 0A 49 10 26 01 30 86 08 97 49 96 4E 90 26 97 4E :A7
52A0 10 24 01 22 9E 4A 63 84 63 89 40 00 63 88 50 63 :F0
52B0 89 40 50 63 89 00 A0 63 89 40 A0 30 01 9F 4A 0C :97
52C0 4D 16 01 02 B6 D3 60 D6 26 0D F3 26 20 4D 10 27 :15
52D0 01 CA 17 FE CF 86 A8 C6 34 17 01 7A 17 00 B2 86 :B8
52E0 80 B7 D3 FF B6 D4 04 B6 D4 04 7E BE 80 81 35 26 :BD
52F0 34 C1 0A 10 27 00 9A 0A FD 26 1D 86 03 97 FD 0C :43
-----------------------------------------------------------
Sum: 0A 75 45 27 B5 C2 EC 9A 92 7F FD 44 28 7F B9 4D :E7

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5300 26 0C 37 0C 3B 0C 3F 96 1F 8B 06 97 1F 96 26 B7 :6A
5310 D3 FF B6 D4 04 B6 D4 04 DC 33 17 01 39 DC 33 80 :DD
5320 03 97 33 20 6C 81 32 26 32 C1 01 27 64 0A FE 26 :DF
5330 1D 86 03 97 FE 0A 26 0A 37 0A 3B 0A 3F 96 1F 80 :6F
5340 06 97 1F 96 26 B7 D3 FF B6 D4 04 B6 D4 04 DC 33 :2C
5350 17 01 03 DC 33 8B 03 97 33 20 36 81 33 26 18 DC :A6
5360 33 17 00 F2 A6 01 43 27 28 A6 89 F9 71 43 27 21 :99
5370 DC 33 5C DD 33 20 1A 81 31 26 16 DC 33 17 00 D6 :9F
5380 A6 1F 43 27 0C A6 89 F9 6F 43 27 05 DC 33 5A DD :87
5390 33 DC 33 17 00 7F 9E 1A CE D2 A5 86 14 97 19 C6 :E5
53A0 50 A6 C0 43 A7 84 3A 0A 19 26 F6 30 89 39 C0 CE :1D
53B0 D2 A5 86 14 97 19 A6 C0 A7 84 3A 0A 19 26 F7 96 :62
53C0 F3 10 26 00 D7 39 DC 35 27 14 17 00 98 DC 35 9B :E0
53D0 37 DB 38 DD 35 8D 3E 4D 27 04 0F 35 0F 36 DC 39 :3D
53E0 27 14 17 00 80 DC 39 9B 3B DB 3C DD 39 8D 26 4D :EA
53F0 27 04 0F 39 0F 3A DC 3D 10 27 03 10 17 00 66 96 :32
-----------------------------------------------------------
Sum: B8 53 E1 83 C0 4E D4 3F 3C 22 93 BC 2F 5E 58 A1 :C3

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5400 3D 9B 3F 97 3D D6 3E 8D 0C 4D 10 27 02 FE 0F 3D :68
5410 0F 3E 16 02 F7 34 04 C6 50 3D 1F 01 35 04 3A 9F :19
5420 1A 1F 12 31 A9 80 00 CE D2 69 86 14 97 19 C6 50 :0E
5430 8C 06 40 25 16 8C 3E 80 24 11 A6 C4 A7 1F A7 3F :A2
5440 A6 41 A7 A4 A6 42 A7 01 A7 21 4F 3A 31 A8 50 33 :6F
5450 43 0A 19 26 DB 39 8D 0D 30 89 39 C0 86 14 6F 84 :79
5460 3A 4A 26 FA 39 34 04 C6 50 3D 1F 01 35 04 3A 1F :1A
5470 13 33 C9 80 00 86 14 97 19 86 FF C6 50 8C 06 40 :46
5480 25 11 8C 3E 80 24 0C A7 1F A7 5F A7 C4 A7 84 A7 :B9
5490 01 A7 41 3A 33 C8 50 0A 19 26 E2 39 DC 35 10 27 :1A
54A0 00 B6 96 37 26 0C 0A FA 26 08 86 14 97 FA 0A 37 :53
54B0 0A 37 96 32 26 4E D6 39 27 3E 96 37 2A 2A CB 28 :05
54C0 D1 35 25 34 C0 3C D1 35 24 2E 96 36 C6 01 91 3A :11
54D0 27 13 24 08 8B 02 91 3A 24 64 20 1C 5F 80 03 91 :F5
54E0 3A 25 5B 20 13 5F 20 56 D6 35 C0 14 D1 39 24 08 :D7
54F0 CB 3C D1 39 25 02 20 D2 D6 30 96 17 84 40 27 26 :EE
-----------------------------------------------------------
Sum: 55 14 C4 A9 2F 30 AA 87 0B 7B 6A 69 8C 80 FD A7 :6F

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5500 0F 38 20 6B 96 36 D6 3A 26 0C 81 17 27 F2 81 1C :2E
5510 27 EE C6 1C 20 08 C6 1C 91 3A 24 02 C6 17 D1 36 :D6
5520 27 DE 24 2E 20 22 96 38 27 10 96 36 81 12 27 D0 :F4
5530 81 21 27 CC 96 37 27 C8 20 35 96 37 27 31 96 36 :97
5540 C4 01 27 0A 81 12 27 B8 86 FF 97 38 20 21 81 21 :9F
5550 27 AE 86 01 97 38 20 17 96 16 84 01 27 11 0F 38 :12
5560 8E 7F 39 CE 7F 3D 17 01 19 DD 35 96 1A 97 37 DC :6D
5570 39 10 27 00 B3 96 3B 26 0C 0A FB 26 08 86 14 97 :8A
5580 FB 0A 3B 0A 3B 96 32 26 4B D6 35 27 3B 96 3B 2A :26
5590 27 CB 28 D1 39 25 31 C0 3C D1 39 24 2B 96 3A C6 :65
55A0 01 91 36 27 69 24 08 8B 02 91 36 24 61 20 19 5F :F5
55B0 80 03 91 36 25 58 20 10 D6 39 C0 14 D1 35 24 08 :0C
55C0 CB 3C D1 35 25 02 20 D5 D6 30 96 17 84 40 26 26 :EC
55D0 0F 3C 20 70 96 3A D6 36 26 0A 81 17 27 F2 81 1C :35
55E0 27 EE 20 08 C6 1C 91 36 27 02 24 02 C6 17 D1 3A :1D
55F0 27 DE 24 2E 20 22 96 3C 27 10 96 3A 81 12 27 D0 :FC
-----------------------------------------------------------
Sum: 5B 10 9D 6D 59 65 9A 4A EB 44 51 68 88 77 3B C7 :FD

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5600 81 21 27 CC 96 3B 27 C8 20 3A 96 3B 27 36 96 3A :AD
5610 C4 01 27 0A 81 12 27 B8 86 FF 97 3C 20 26 81 21 :A8
5620 27 AE 86 01 97 3C 20 1C 96 17 84 80 26 16 0F 3C :A3
5630 8E 7F 35 CE 7F 3D 8D 4A DD 39 96 1A 2A 02 80 04 :19
5640 8B 02 97 3B DC 3D 27 22 96 3E 91 36 26 07 D6 37 :96
5650 D7 3F 16 00 B7 91 3A 26 04 D6 3B 20 F3 96 3F 10 :E1
5660 26 00 A9 0A 3F 0A 3F 16 00 A2 96 25 4A 10 26 00 :54
5670 9B 8E 7F 35 CE 7F 39 8D 09 DD 3D 96 1A 97 3F 16 :AF
5680 00 8A 96 26 81 06 24 09 86 C8 C6 05 D0 26 53 20 :7C
5690 05 4F D6 26 C0 05 97 18 D7 1A 96 32 26 3A D6 30 :E3
56A0 C4 0F CB 12 D7 19 A6 84 27 29 D6 18 2B 06 81 28 :E2
56B0 24 21 20 04 81 A0 25 1B 96 19 8B 03 81 19 24 13 :D8
56C0 80 05 25 0F 4F 5F 39 11 83 7F 3D 26 06 96 19 A1 :6C
56D0 01 27 F1 96 18 D6 19 39 86 17 D6 30 C4 01 27 02 :80
56E0 86 1C 97 19 96 18 2B 12 EC 84 27 04 81 32 25 D4 :84
56F0 EC C4 27 D3 81 32 25 CC 20 CD EC 84 27 04 81 96 :ED
-----------------------------------------------------------
Sum: FD 33 09 12 E4 60 FC B9 EB 27 C9 52 28 04 D4 90 :01

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5700 24 C2 EC C4 27 C1 81 96 24 BA 20 BB 96 FF 84 01 :68
5710 10 26 00 AC 96 33 C6 50 3D 1F 01 D6 34 3A 30 89 :1B
5720 7F B0 A6 84 43 26 28 30 89 01 40 A6 1F 43 26 1F :31
5730 A6 01 43 26 1A 30 89 04 60 A6 1F 43 26 11 A6 01 :2D
5740 43 26 0C A6 89 00 F0 43 26 05 0F F2 16 00 71 96 :20
575Ø F3 1Ø 27 ØØ 6B 96 F2 ØC F2 4D 26 ØE 86 ØC B7 D3 :B8
576Ø FF B6 D4 Ø4 B6 D4 Ø4 16 ØØ 56 81 Ø1 1Ø 26 ØØ 5Ø :8F
577Ø 86 ØD B7 D3 FF B6 D4 Ø4 B6 D4 Ø4 86 14 F7 D4 Ø8 :A5
578Ø 8E ØB B8 3Ø 1F 26 FC F6 D4 Ø8 8E ØB B8 3Ø 1F 26 :5A
579Ø FC 4A 26 E9 96 F1 81 2B 27 1Ø 86 Ø3 1Ø 8E ØØ ØØ :E6
57AØ 31 3F 26 FC 4A 26 F5 16 FA 6B 86 81 B7 D3 FF 96 :98
57BØ 4D D6 4E 53 FD D3 FD B6 D4 Ø4 B6 D4 Ø4 7E BE 8Ø :69
57CØ 96 26 1Ø 27 FA 88 96 31 27 5D 4A 26 3Ø ØC 31 8E :2B
```

INDY-7　マシン語プログラム・リスト

```
57D0 09 78 10 8E 89 78 33 8D 02 9B 86 14 97 19 EC C1 :74
57E0 A7 84 A7 A4 E7 03 E7 23 30 88 50 31 A8 50 0A 19 :BE
57F0 26 EC 86 0E B7 D3 FF B6 D4 04 B6 D4 04 0A 32 10 :97
------------------------------------------------------------
Sum: 88 0A 32 66 E0 50 D0 07 0E 07 60 A3 C5 44 B1 1F :22


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5800 26 00 81 0C 25 0F 31 0C 32 8E 09 78 10 8E 89 78 :04
5810 86 14 C6 FF E7 84 E7 03 E7 A4 E7 23 30 88 50 31 :82
5820 A8 50 4A 26 EF 20 5D 96 25 26 2E 96 2F 26 55 0A :2D
5830 F6 26 51 0A F7 26 4D 96 30 8A 01 84 03 97 F7 0C :53
5840 31 86 32 97 32 0F F9 96 30 26 39 86 01 97 F7 96 :8A
5850 17 8A 01 84 3F 97 F6 20 2B 4A 26 28 96 F9 27 08 :93
5860 96 3D 26 20 0C 25 20 1C 96 2B 81 B4 26 16 0A F5 :B7
5870 26 12 0A F8 26 0E 96 30 8A 01 84 03 97 F8 96 17 :82
5880 97 F5 0C F9 8E 86 4A 10 8E 46 50 96 26 9B 28 97 :39
5890 28 86 B4 97 19 96 25 26 10 8D 6C CC 06 50 DD 29 :24
58A0 0F 2B DD 2D 0F 2F 16 F9 A5 4A 26 35 DE 29 1F 31 :32
58B0 30 89 7F FA 1F 32 31 A9 40 00 17 00 E3 8E 86 4A :F5
58C0 10 8E 46 50 CE 06 50 96 2B 97 1A 86 B4 90 1A 97 :45
58D0 19 17 00 A0 96 19 10 27 F9 74 DE 20 8D 29 16 F9 :E6
58E0 6D DE 2D 17 01 31 8E 86 4A 10 8E 46 50 96 2F 97 :AF
58F0 19 86 B4 90 19 97 1A 8D 0E 96 25 10 27 F9 4F DE :60
------------------------------------------------------------
Sum: FB 21 88 BC E8 16 25 E5 E8 4C 27 AD 6B 5B 3B A9 :1A


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5900 2D 17 00 70 16 F9 47 DC 20 93 1E 10 93 22 24 0D :AD
5910 DD 1C DC 22 93 1C DD 1C CC D2 68 93 1C 1F 03 96 :0C
5920 25 4A 27 02 DF 20 EC C4 2A 04 DE 22 EC C4 ED 84 :96
5930 ED 88 1C EC 42 ED 02 ED 88 1E EC 44 ED 04 ED 88 :D7
5940 1A 0A 28 26 08 86 05 97 24 86 3C 97 28 0A 24 2A :99
5950 0A CC F8 1F A7 06 E7 88 19 20 05 6F 06 6F 88 19 :CC
5960 86 FF A7 A4 A7 A8 13 30 88 50 31 A8 50 33 46 0A :E6
5970 19 26 B3 39 0A 28 26 08 86 05 97 24 86 3C 97 28 :52
5980 86 81 0A 24 2A 02 86 E7 A7 0B A7 88 14 43 A7 45 :F2
5990 A7 4E 30 88 50 31 A8 50 33 C8 50 0A 1A 26 D5 39 :C9
59A0 11 83 3E 90 27 69 96 26 97 1A 9B 2B 97 2B CC 00 :B3
59B0 00 ED 84 ED 02 ED 04 ED 06 ED 08 A7 0A 34 10 30 :5E
59C0 88 14 E7 01 ED 02 ED 04 ED 06 ED 08 ED 0A 35 10 :88
59D0 6F A4 6F A8 13 CC FF FF ED 5A ED 5C ED 5E ED C4 :93
59E0 ED 42 A7 44 E7 25 A7 2E 34 40 33 4E E7 41 ED 42 :47
59F0 ED 44 ED 46 ED 48 ED 4A 35 40 30 88 50 31 A8 50 :76
------------------------------------------------------------
Sum: EE 7D 7F FE A1 42 7F C5 A3 3C 30 79 6C 93 99 38 :67


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5A00 33 C8 50 11 83 3E 90 27 07 0A 1A 26 A1 DF 29 39 :07
5A10 DF 29 86 B4 97 2B 39 96 26 97 1A 9B 2F 97 2F 1F :59
5A20 31 30 89 80 07 1F 32 31 A9 40 05 CC 00 00 ED 5A :F4
5A30 ED 5C ED 5E 33 C8 14 ED C4 ED 42 ED 44 CC FF FF :7E
5A40 A7 5A A7 51 ED 19 ED 1B ED 1D ED 07 ED 09 ED 0B :F3
5A50 6F A4 6F 29 30 88 50 31 A8 50 33 C8 3C 11 83 3E :E5
5A60 90 27 07 0A 1A 26 C4 DF 2D 39 DF 2D 86 B4 97 2F :1D
5A70 0F 25 0F 32 39 3F FC CF F3 F3 CF FC 3F FC 3F FC :DF
5A80 3F FC 3F FC 3F FC 3F FC 3F FC 3F FC 3F FC 3F FC :D8
5A90 3F FC 3F FC 3F FC 3F F3 CF CF F3 3F FC 00 01 80 :30
5AA0 00 C2 63 00 03 C0 00 31 8C 00 0F F0 00 0C B0 00 :60
5AB0 3F FC 00 02 C0 00 FF FF 00 01 80 00 0F F0 00 01 :7C
5AC0 80 00 1F F8 00 00 00 00 7F FE 00 00 00 01 FF FF :13
5AD0 80 00 00 07 FF FF E0 00 00 00 1F FC 00 C0 03 00 :43
5AE0 3F FE 00 30 0C 00 7F FF 00 08 10 01 FF FF C0 04 :D2
5AF0 20 07 FF FF F0 C4 23 1F FF FF FC 32 4C 7F C1 83 :56
------------------------------------------------------------
Sum: 01 82 77 81 00 D1 0B 12 67 38 35 CC 97 43 FD 28 :08


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5B00 FF 0A 50 78 01 80 0F C5 A3 00 03 C0 00 33 CC 00 :8B
5B10 07 E0 00 0F F0 00 3C 00 00 00 E0 00 7E 00 00 00 :80
5B20 00 00 3C 00 00 00 00 00 FF 00 00 07 E0 01 FF 80 :A2
5B30 00 1F F8 03 FF C0 00 7F FE 01 FF 80 00 7F FE 07 :5A
5B40 FF E0 00 FF FF 1F FF F8 00 7F FE 7F 18 FE 00 1F :24
5B50 F8 00 18 00 00 01 80 00 18 03 C0 03 C0 00 3C 07 :72
5B60 E0 07 E0 00 FF 0F F0 0F F0 06 00 07 E0 00 00 0F :C0
5B70 00 0F F0 C0 78 1F 80 1F F8 20 FC 1F 80 3F FC 11 :F4
5B80 FE 3F C0 0F F0 09 FE 3F C0 1F F8 05 FE 3F C0 3F :5A
5B90 FC 65 FE 3F C0 7F FE 15 FE 7F E0 7F FE 0D FE 7F :54
5BA0 E0 7F FE 1D FE 7F E0 3F FC 1D FE 7F E0 01 80 00 :0D
5BB0 FC 3F C0 03 C0 00 78 1F 80 0F F0 E0 30 0F 00 00 :F3
5BC0 00 18 78 06 00 7F 00 04 FC 06 01 FF C0 C4 00 06 :A5
5BD0 03 FF E0 32 30 06 01 FF C0 0A 47 0F 07 FF F8 09 :71
5BE0 48 3F C3 FF F0 05 90 00 0F FF FC 07 20 00 3F FF :3D
5BF0 FF 03 C0 00 00 0C 00 03 E0 00 00 1E 00 00 00 00 :CF
------------------------------------------------------------
Sum: FD BA C3 EE F4 2B 1F 22 85 82 A6 05 89 0F 76 99 :21


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5C00 00 3F 80 00 00 80 FF E7 FF FF C3 FF FF C3 FF FF :A5
5C10 C3 FF E0 81 07 C0 00 03 E0 81 07 FF 99 FF FF A5 :90
5C20 FF FF A5 FF FF 24 FF FF 42 FF 81 42 81 00 42 00 :8A
5C30 00 42 00 00 42 00 81 3C 81 FF 00 FF FF 00 FF FF :BD
5C40 81 FF 18 3C 3C 3C 7E FF 7E 7E 7E 7E FF FF FF FF :BD
5C50 FF FF FF FF FF 7E 00 00 00 00 00 00 00 00 00 00 :79
------------------------------------------------------------
Sum: 42 7D 1C BB 83 1E FD 24 20 FC C9 BD 17 C1 3E A2 :B2
```

FM7/11+Kコンパイラ

メロン・パニック

■三浦　貢

どこかの国のどこかの村．天気も良いのでファーマーさん，心うきうき胸弾ませて，今日も元気に農作業．手塩にかけたバナナにリンゴ，ブドウにメロンにトマトにエトセトラ．雨土御代のお恵みと思いきや，突然メロンが化物に…

ゲームについて

上下左右それぞれ8246のキーを使ってファーマーさんを動かしてください．もし進行方向に干草の束があれば（黄色の四角）それも同方向に動きます．つまり，ファーマーさんが干草を押す格好になるわけです．

この干草でメロンをつぶせば良いのですがフルーツと干草の間にメロンを挟んでつぶすことはできません．干草の中には，ボーナス・ブロックが入っています．これをそろえると200点がプラスされ，フルーツもその種類に応じて点数が加えられます．ただし，メロンがフルーツを食べた場合はマイナスとなります．

画面中，TNTと書かれているのはダイナマイトの点火スイッチなので，ファーマーさんはもちろんのこと，メロン達にも触わらせないよう注意してください．

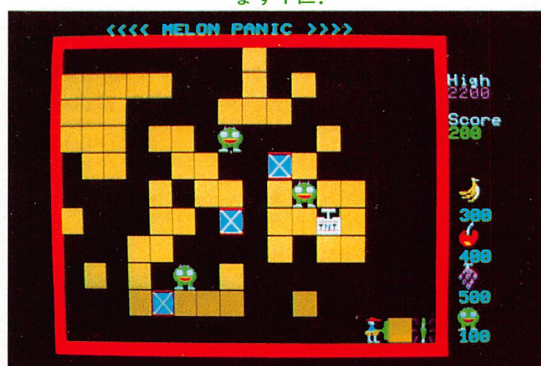
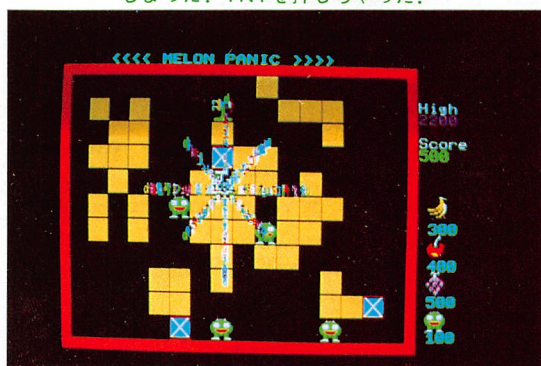
メロンをすべてつぶせば5000点プラスされ，面クリアとなります．しかし，メロンはつぶされても，しばらくすると再び現われるので，全部のメロンを殺すには一度にすばやくやらなければなりません．これがこのゲームの難しいところです．パニック状態に落ち入らぬよう慎重にそして計画的に，メロンをつぶしてください．

まず1匹．

しまった！TNTを押しちゃった．

プログラムについて

プログラムは，FM-7用K言語で書かれています（Kがわからない人はFM-8活用研究を参照してください）．

データ部分は，＄701B～＄78FFがキャラクタ・データ，＄7900～＄7A30が音楽データとなっています．またコンパイル後のオブジェクトは＄701Bで終わるように作ってあるので，ソースの改造はしないほうがよいと思います．もっともデータ部分をアドレスの低いほうに移せば，改造も可能となります．しかし，ランタイム・ルーチンとの関係もあって，オブジェクトがきわめて長くなってしまうと思います．このプログラムの場合，データはすべて高アドレスに置かれています．

コンパイル後は，＄4D00～＄7A30を**SAVEM**してください．スタート・アドレスは，いつも通り＄5000です．

◆参考文献

FM-8活用研究

MELON PANIC Kコンパイラ用ソース・リスト

```
10 'OTDT=$7900;BBB=OTDT+$110
20 'INIT[]
30 'AS=$3000;P=AS+250;Q=P+12;P(0)=0;Q(0)=0;P(1)=
-32;Q(1)=0;P(2)=0;Q(2)=16;P(3)=3
2;Q(3)=0;P(4)=0;Q(4)=-16
40 'PP=Q+12;QQ=PP+12;PP(0)=0;QQ(0)=0;PP(1)=-1;QQ
(1)=0;PP(2)=0;QQ(2)=17;PP(3)=1;Q
Q(3)=0;PP(4)=0;QQ(4)=-17;HSC=0;SC=0
50 'STT;WID[];LOC[35,4,7];PRINT"High";LOC[35,5,3
];PRINTHSC;LOC[35,7,7];PRINT"Sco
re";SCR[0]
60 'LIN[2,135,1,1,34,24,1]
70 'LOC[5,0,5];PRINT" <<<< MELON PANIC >>>>"
80 'FOR L=5TO8;Y=90+(L-5)*25
90 'PUT[576,Y+4,590,Y+18,591,605,L]
100 'LOC[36,Y/8+3,5];?(L-2)*100
110 'NEXT;LOC[36,Y/8+3,5];?100
120 'FOR L=18TO216;AS:L)=0;NEXT
130 'FOR L=0TO17;AS:L)=1;NEXT
140 'FOR L=0TO17;AS:$CC+L)=1;NEXT
150 'FOR L=0TO12;AS:17*L)=1;NEXT
160 'FOR L=0TO12;AS:$11+17*L)=1;NEXT
170 'FOR L=1TO65;X=RND(12)+2;Y=RND(9)+2
180 'AS:X+17*Y)=2
190 'LINE[6,X*32,Y*16,X*32+30,Y*16+14,2];NEXT
200 'X=RND(9)+5;Y=RND(7)+4
210 'X1=X*32+16;Y1=Y*16+8
220 'PUT[X*32,Y*16,X*32+14,Y*16+14,X*32+15,X*32+
29,4]
230 'LINE[6,(X-1)*32,Y*16,(X-1)*32+30,Y*16+14,2]
240 'LINE[6,(X+1)*32,Y*16,(X+1)*32+30,Y*16+14,2]
250 'LINE[6,X*32,(Y-1)*16,X*32+30,(Y-1)*16+14,2]
260 'LINE[6,X*32,(Y+1)*16,X*32+30,(Y+1)*16+14,2]
270 'AS:X+17*Y)=4
280 'AS:X-1+17*Y)=2;AS:X+17*(Y-1))=2
290 'AS:X+1+17*Y)=2;AS:X+17*(Y+1))=2
300 'FOR L=1TO3
310 'ZZ;X=RND(12)+3;Y=RND(7)+4
320 'IF AS:X+17*y)=5 THEN GOTO ZZ;FI
330 'AS:X+17*Y)=5
340 'A=X*32;B=Y*16;R1[];NEXT
350 'U=$3300;W=U+25;CU=W+25;CW=CU+25;MT=CW+25;MF
=MT+25;MA=MF+25;E44=0
360 'FOR L=1TO4;MF(L)=1
370 'Z11[];NEXT
390 'X=32;Y=16;CX=X;CY=Y;T=0;AD=18;T1=0
400 'Z1;
410 'PSG[12,1];PSG[6,1];PSG1[BBB:E44),BBB:E44+1)
,$F8];E44=E44+2;IF BBB:E44)=255
THEN E44=0;FI
420 'INK[];IF KY=52THEN T=1;T1=0;FI
430 'IF KY=54THEN T=3;T1=1;FI
440 'IF KY=50THEN T=2;T1=3;FI
450 'IF KY=56THEN T=4;T1=2;FI
460 'E=0;AC=0;I1=0;Z2[]
470 'AD=AD+PP(T)+QQ(T)
480 'X=X+P(T);Y=Y+Q(T)
490 'LINE[0,CX,CY,CX+30,CY+14,2]
500 'CX=X;CY=Y
510 'PUT[X,Y,X+14,Y+14,X+15,X+29,T1]
520 'G1=0;FOR L=1TO4
530 'IF MF(L)>1THEN MF(L)=MF(L)-1;IF MF(L)=1THEN
 Z11[];G1=G1-1;FI;G1=G1+1;GOTO N
;FI;Z3[]
540 'IF X=U(L)THEN IF Y=W(L)THEN GOTO EN;FI;FI
550 'N;NEXT
560 'IF G1=4 THEN GOTO EN2;FI
570 'IF RND(30)=1 THEN FR[];FI
580 'AS:AD)=0
590 'IF AC<>0THEN Z10[];FI
600 'GOTO Z1
610 'Z2;E=E+1
620 'I=AS:AD+E*(PP(T)+QQ(T)))
630 'IF I=1THEN T=0;RETURN;FI
640 'IF I=0THEN GOTO R2;FI
650 'IF I=3THEN Z4[];GOTO R2;FI
660 'IF I=4THEN GOTO BAN;FI
670 'IF I=5THEN AC=AD+(E+1)*(PP(T)+QQ(T));FI
680 'IF I>5THEN IF E=1THEN SCR[(I-3)*100];MUS[2]
;FI;AS:AD+E*(PP(T)+QQ(T)))=0;GOT
O R2;FI
690 'GOTO Z2
700 'R2;I=AS:AD+(E-1)*(PP(T)+QQ(T)));AS:AD+E*(PP
(T)+QQ(T)))=I
710 'I0=I1-I;I1=I;E=E-1
720 'IF I0=0THEN GOTO R3;FI
730 'IF ABS(I0)=2 THEN Z8[];GOTO R3;FI
740 'IF I0=3THEN Z8[];GOTO R3;FI
750 'IF ABS(I0)=5 THEN Z9[];FI
760 'IF I0=-3 THEN Z9[];FI
770 'R3;IF E<1THEN RETURN;ELSE GOTO R2;FI
780 'Z3;IF RND(10)>7THEN IF Y>W(L) THEN MT(L)=2
ELSE MT(L)=4;FI;GOTO R5;fi
790 'IF RND(10)<4THEN IF X>U(L) THEN MT(L)=3 ELS
E MT(L)=1;FI;GOTO R5;fi
800 'MT(L)=RND(4)
810 'R5;I=AS:MA(L)+PP(MT(L))+QQ(MT(L)))
820 'IF I=4 THEN GOTO BAN;FI
830 'IF I>5 THEN SCR[(3-I)*100];MUS[5];GOTO Z7;F
I
840 'IF I<>0THEN MT(L)=0;FI
850 'GOTO Z7
860 'Z4;IF E=1THEN LINE[0,CX,CY,CX+30,CY+14,2];G
OTO EN;FI
870 'FOR L=1TO4;IF MA(L)=AD+E*(PP(T)+QQ(T)) THEN
 GOTO Z12;FI;NEXT
880 'Z12;I=AS:MA(L)+PP(T)+QQ(T))
890 'IF I=4 THEN GOTO BAN;FI
900 'IF I>5 THEN GOTO Z7;FI
910 'IF I<>0THEN SCR[100];GOTO MD;FI;MT(L)=T
920 'Z7;CU(L)=U(L);CW(L)=W(L)
930 'U(L)=U(L)+P(MT(L));W(L)=W(L)+Q(MT(L))
940 'LINE[0,CU(L),CW(L),CU(L)+30,CW(L)+14,2]
950 'PUT[U(L),W(L),U(L)+14,W(L)+14,U(L)+15,U(L)+
29,8]
960 'AS:MA(L))=0
970 'MA(L)=MA(L)+PP(MT(L))+QQ(MT(L))
980 'AS:MA(L))=3;RETURN
990 'MD;LINE[0,U(L),W(L),U(L)+30,W(L)+15,2];IF T
1<2THEN TJ=9;ELSE TJ=11;FI
1000 'PUT[U(L),W(L),U(L)+14,W(L)+14,U(L)+15,U(L)
+29,TJ]
1010 'PUT[U(L),W(L),U(L)+14,W(L)+14,U(L)+15,U(L)
+29,TJ]
1020 'LINE[0,U(L),W(L),U(L)+30,W(L)+15,2]
1030 'PUT[U(L),W(L),U(L)+14,W(L)+14,U(L)+15,U(L)
+29,TJ+1]
1040 'PSG[1,10];MUS[1]
1050 'AS:MA(L))=0;MF(L)=20;RETURN
1060 'Z11;U(L)=(RND(10)+2)*32;W(L)=(rnd(9)+2)*16
;MA(L)=U(L)/32+W(L)/16*17;CW(L)=
W(L);CU(L)=U(L)
1070 'I=AS:MA(L));IF I<>0 THEN GOTO Z11;FI
1080 'AS:MA(L))=3
1090 'PUT[U(L),W(L),U(L)+14,W(L)+14,U(L)+15,U(L)
+29,8];RETURN
1100 'Z8;Z13[]
1110 'LINE[6,A,B,A+30,B+14,2];RETURN
1120 'Z9;Z13[]
1130 'R1;LINE[1,A,B,A+30,B+14,2]
1140 'LINE[6,A,B,A+30,B+14,0]
1150 'LINE[6,A+30,B,A,B+14,0]
1160 'LINE[2,A,B,A+30,B+14,1]
1170 'RETURN
1180 'Z10;IF AS:AC)<>5THEN RETURN;FI
1190 'IF AS:AC-1)<>5THEN GOTO R7;FI
1200 'IF AS:AC-2)=5THEN GOTO EN1;FI
1210 'IF AS:AC+1)=5THEN GOTO EN1;FI
1220 'RETURN
1230 'R7;IF AS:AC-17)<>5THEN GOTO R8;FI
1240 'IF AS:AC-34)=5 THEN GOTO EN1;FI
1250 'IF AS:AC+17)=5 THEN GOTO EN1;FI
1260 'RETURN
1270 'R8;IF AS:AC+1)<>5THEN GOTO R9;FI
1280 'IF AS:AC+2)=5 THEN GOTO EN1;FI
1290 'RETURN
1300 'R9;IF AS:AC+17)<>5THEN RETURN;FI
1310 'IF AS:AC+34)=5 THEN GOTO EN1;FI
1320 'RETURN
1330 'Z13;PSG1[50,0,$DC];A=X+P(T)*(E+1);B=Y+Q(T)
*(E+1);RETURN
1340 'FR;PSG1[70,180,$F8];FX=RND(14)+1;FY=RND(10
)+1
1350 'IF AS:FX+FY*17)<>0THEN RETURN;FI
1360 'IF RND(4)=0THEN F1=7;GOTO FS;FI
1370 'IF RND(4)=1THEN F1=6;GOTO FS;FI
1380 '  F1=5
1390 'FS;AS:FX+FY*17)=F1+1
1400 'FX=FX*32;FY=FY*16
1410 'PUT[FX,FY,FX+14,FY+14,FX+15,FX+29,F1]
1420 'RETURN
1430 'SCR;SC=SC+%1;LOC[35,8,4];?SC
1440 '  IF SC>HSC THEN HSC=SC;LOC[35,5,3];?HSC;FI
;RETURN
```

MELON PANIC Kコンパイラ用ソース・リスト

```
1450 'EN1;MUS[8];SCR[2000];FOR L=18TO216;IF AS:L
)=5THEN AS:L)=2;FI;NEXT;RETURN
1460 'EN2;FOR L=1TO500;PSG1[50,0,$F8];SCR[10];PS
G1[0,0,$F8];NEXT;MUS[7];GOTO STT

1470 'EN;MUS[6]
1480 'ENE;MUS[9];REPEAT;INK[];IF KY='E'THEN ENDF
I;UNTIL KY='S';SC=0;GOTO STT
1490 'SY;HLT[];ADDR=$FC82;CPOKE ADDR ,$19,%1,0,0
,1,1,%2#,%3#,1,%4;SUB[];RETURN
1500 'SYM;HLT[];ADDR=$FC82;CPOKE ADDR ,$19,%1,0,
0,%2,%3,%4#,%5#,8,%7,%8,%9,%10,%
11,%12,%13,%14;SUB[];RETURN
1510 'WID;HLT[];ADDR=$FC82;CPOKE ADDR ,$1,0,40,2
5,0,25,0,1,0;SUB[];RETURN
1520 'LOC;HLT[];ADDR=$FC82;CPOKE ADDR ,$3,5,$12,
%1,%2,$11,%3;SUB[];RETURN
1530 'LIN;HLT[];ADDR=$FC82;CPOKE ADDR,$20,%1,%2,
%3#,%4#,%5#,%6#,%7;SUB[];RETURN
1540 'HLT;CODE $B6,$FD,5,$2B,$FB,$8680#,$B7,$FD,
5,$B6,$FD,5,$2A,$FB,$39
1550 'SUB;CODE $7F,$FD,5,$39
1560 'LINE;HLT[];ADDR=$FC82;CPOKE ADDR,$15,%1,0,
%2#,%3#,%4#,%5#,%6;SUB[];RETURN
1570 'INK;HLT[];ADDR=$FC82;CPOKE ADDR,$29,0;SUB[
];HLT[];KY=PEEK($FC83);POKE $FC8
2,$3F;SUB[];RETURN
1580 'PUT;A1=$701B+175*%7;HLT[];ADDR=$FC82;CPOKE
 ADDR,$1E,%1#,%2#,%3#,%4#,0,0,87
;FOR N0=0TO87;ADDR:N0)=A1:N0);NEXT;SUB[];RETURN
1590 'HLT[];ADDR=$FC82;CPOKE ADDR,$1E,%5#,%2#,%6
#,%4#,0,0,87;FOR n0=88TO175;ADDR
:N0-88)=A1:n0);NEXT;SUB[];RETURN
1600 'BAN;INIT[];PSG[1,$1F];PSG[12,$60];PSG1[0,0
,$DC];LINE[0,CX,CY,CX+30,CY+14,2
]
1610 'PUT[X,Y,X+14,Y+14,X+15,X+29,T1]
1620 'FOR J1=0TO30;BAX1=X1+J1*4;BAX2=X1-J1*4;BAX
3=X1+J1*2;BAX4=X1-J1*2
1630 'BAY1=Y1+J1*2;BAY2=Y1-J1*2;BAY3=Y1+J1;BAY4=
Y1-J1;FD=RND(200)+100;BC=RND(5)
1640 'SY[7,BAX1,y1,FD];SY[4,BAX2,Y1,FD];SY[5,X1,
BAY1,FD];SY[2,X1,BAY2,FD];SY[3,B
AX3,BAY3,FD];SY[2,BAX4,BAY4,FD];SY[3,BAX3,BAY4,F
D];SY[2,BAX4,BAY3,FD]
1650 'SY[BC,BAX1,y1,FD];SY[BC+2,BAX2,Y1,FD];SY[B
C+1,X1,BAY1,FD];SY[BC,X1,BAY2,FD
];SY[BC,BAX3,BAY3,FD];SY[BC,BAX4,BAY4,FD];SY[BC+
1,BAX3,BAY4,FD];SY[BC,BAX4,BAY3,
FD]
1660 'NEXT;INIT[];GOTO ENE
1670 'MUS;PSG[12,10]
1680 'IF %1=1THENSND=0;WAI=3;FI
1690 'IF %1=2THENSND=$20;WAI=3;FI
1700 'IF %1=4THENSND=$30;WAI=3;FI
1710 'IF %1=5THENSND=$45;WAI=3;FI
1720 'IF %1=6THENSND=$60;WAI=6;PSG[1,3];psg[3,1]
;FI
1730 'IF %1=8THENSND=$70;WAI=8;FI
1740 'IF %1=9THENSND=$E0;WAI=8;FI
1750 'IF %1=7THENSND=$90;WAI=8;FI
1760 'SND1=SND+OTDT
1770 'ONGA[SND1,WAI];INIT[];RETURN
1780 'ONGA;SOUN=%1;REPEAT
1790 'PSG1[SOUN:1),SOUN:2),$F8]
1800 'FOR LL1=1 TO SOUN:0);FOR LL=0 TO %2*500;NE
XT;NEXT;SOUN=SOUN+3
1810 'UNTIL SOUN:0)=0;RETURN
1820 'INIT;PSG[6,$1F];PSG[7,$F8];PSG[8,$1F];PSG[
9,$1F];PSG[10,$1F];PSG[12,10]
1830 'FOR LL=0 TO 5;PSG[LL,0];NEXT;RETURN
1840 'PSG1;PSG[7,%3];PSG[0,%1];PSG[2,%2];PSG[13,
3];RETURN
1850 'PSG;POKE$FD0E,%1;POKE$FD0D,3;POKE$FD0D,0;P
OKE$FD0E,%2;POKE$FD0D,2;POKE$FD0
D,0;RETURN
```

キャラクタ＆音楽データ ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7010 00 00 00 00 00 00 00 00 00 00 00 4F F0 00 00 00 :3F
7020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7030 00 00 00 00 00 03 F0 00 00 00 7C 00 00 00 00 00 :6F
7040 00 00 00 00 00 00 00 00 02 00 1C 00 38 00 10 00 :66
7050 00 00 00 03 F0 00 00 00 00 00 07 80 0F 00 1E 00 :A7
7060 3E 00 7F FF F8 03 E0 1F C0 3F 80 1F 00 00 00 00 :54
7070 03 F0 00 00 00 00 00 3F 81 FC 00 60 03 F8 1F F8 :21
7080 0F F0 00 00 00 00 00 00 00 06 01 8C 03 38 7E 00 :4B
7090 1F 81 FF FC 3F 81 FC 00 60 00 00 00 00 00 FF 00 :B6
70A0 F9 FF FF E7 FF CF C6 01 8C 03 38 7E 00 00 00 00 :B8
70B0 00 3F 81 FC 00 60 00 00 00 00 70 00 FF F9 FF FF :82
70C0 E7 FF CF C6 01 8C 03 38 7E 00 00 00 00 3F 80 :80
70D0 1F 80 0C 00 7E 03 FF 1F E0 00 00 00 00 00 00 60 :8A
70E0 60 C0 C1 81 83 F3 F8 03 F1 FF FC 3F 80 1F 80 0C :29
70F0 00 00 00 00 00 00 00 FF FE 7C FC F8 60 60 C0 C1 :AD
-----------------------------------------------------------
Sum: CE DE 9A 28 28 35 9E A8 7C BF 44 0B 1C A8 49 A3 :4B

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7100 81 83 F3 F8 00 00 00 00 3F 80 1F 80 0C 00 00 00 :59
7110 00 00 1F FF FF FE 7C FC F8 60 60 C0 C1 81 83 F3 :C3
7120 F8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :F8
7130 00 00 00 00 00 00 00 00 00 00 00 3C 00 00 00 00 :3C
7140 01 F0 00 00 00 00 00 00 00 00 00 00 00 E0 01 :D2
7150 C0 02 00 00 00 00 00 00 00 3C 00 00 00 00 00 00 :FE
7160 00 78 00 F0 01 E0 03 C0 1F FF FF E0 7F C0 3E 00 :86
7170 78 00 F0 01 E0 00 3C 00 00 00 00 00 00 00 00 00 :85
7180 00 00 00 00 00 00 01 86 0F 3C 1F FC 3C F8 60 70 :F1
7190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 06 :06
71A0 00 00 0C 00 19 80 3F 01 FF 83 FC 07 E0 03 06 00 :53
71B0 0C 00 00 07 FE 0F FC 00 38 00 10 00 20 00 40 00 :C4
71C0 81 81 0F 02 1F 80 3C 00 60 00 00 00 00 00 00 00 :4E
71D0 00 00 00 00 00 00 00 00 00 00 00 00 00 A0 01 :A1
71E0 40 7F 80 03 00 00 0E 00 10 07 E0 00 00 00 00 00 :47
71F0 00 00 00 00 00 00 00 00 C0 07 E0 9F C1 3F FE 7C :C0
-----------------------------------------------------------
Sum: 7F F9 91 0D 7D BA F5 51 47 3A 94 9B A8 78 E5 E7 :2F

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7200 00 F8 09 FC 17 FF E3 F0 01 80 00 FF C1 FF 83 80 :29
7210 04 00 08 00 10 C0 27 E0 DF C1 BF FE 7C 00 F8 09 :BD
7220 FC 17 FF E3 F0 01 80 00 00 01 F8 1A 00 3C 00 70 :25
7230 00 DF 81 A0 03 40 00 00 00 00 00 00 00 00 00 00 :43
7240 00 00 00 00 00 00 F1 FF E2 0F C4 07 80 0F 1F FE :58
7250 20 FC 40 78 00 00 00 00 00 00 00 00 00 00 00 00 :D4
7260 00 00 00 F1 FF E2 0F C4 07 80 0F 3F FE 60 FC C0 :94
7270 79 00 02 00 04 00 08 00 10 00 E3 FF C7 FF FC C0 :FB
7280 00 00 00 03 81 87 CF 07 FE 0F 3C 18 60 00 00 00 :A2
7290 00 00 00 00 00 00 00 00 00 00 00 00 00 18 00 :18
72A0 30 01 E0 0F F0 7F E0 3F C0 66 00 00 0C 00 18 00 30 :28
72B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 80 :81
72C0 0F 08 7E 10 3C 20 60 40 00 80 01 00 02 00 04 00 :28
72D0 0E 00 1F FE 3F FC 00 03 FE 07 FC 00 08 00 10 00 :82
72E0 3F F0 7F FF FF FF FF FF 81 8F CF 27 9E 7F FF FF :CA
72F0 FF FF FF FC 03 FE 07 FC 00 08 00 10 00 3F F0 7F :C3
-----------------------------------------------------------
Sum: 24 E2 CE 03 0B 01 A7 17 16 64 75 B7 8A 7F AE A5 :A3

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7300 FF FF FF FF FF 81 8F CF 27 9E 7F FF FF FF FF FF :19
7310 FC 03 FE 07 FC 00 08 00 10 00 3F F0 7F FF C0 00 :85
7320 BF FF 01 8E CF 25 9E 78 00 1F FF FF FF FC 0C FF :7A
7330 81 FF 02 00 04 00 08 00 10 7F FF FF FF FF FF FF :17
7340 30 3E 79 F8 F3 FF FF FF FF FF FF 80 FF 81 FF 02 :CD
7350 00 04 00 08 00 10 7F FF FF FF FF FF FF 30 3E 79 :7C
7360 F8 F3 FF FF FF FF FF FF 80 FF 81 FF 02 00 04 00 :EA
7370 08 00 10 7F FF FF FF 80 01 FF FB 30 36 79 E8 F3 D0 :9A
7380 00 3F FF FF FF 80 00 00 00 00 00 00 00 00 00 00 :BC
7390 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
73A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
73B0 00 01 FF FE 7F C0 00 38 7F F0 00 00 00 40 00 00 :24
73C0 00 00 00 00 00 00 00 00 00 00 00 00 00 01 FF :00
73D0 FE 7F C0 00 38 7F F0 00 00 00 40 00 00 00 00 00 :24
73E0 07 E0 00 00 00 00 00 00 01 80 03 80 0F 00 04 00 :FE
73F0 10 00 40 02 00 00 00 00 00 00 1F 00 0F FC 0C :88
-----------------------------------------------------------
Sum: 80 D4 86 11 75 72 2A 7D 44 A4 AE 40 04 E1 FF 53 :86

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7400 00 3E 00 3F 01 FF 8F FF FF FF E7 DE 3F 3F F1 F4 :31
7410 0F 80 7C 1F C0 00 00 00 1F 00 0F FC 0C 00 3E 00 :5E
7420 3F 01 FF 8F FF FF FF E7 DE 3F 3F F1 F4 0F 80 7C :FE
7430 1F C0 00 00 00 00 00 00 00 00 00 00 00 20 00 :FF
7440 C0 01 80 03 00 00 00 00 00 00 00 00 00 00 00 00 :44
7450 00 00 00 00 00 00 00 00 00 00 00 20 00 C1 FF 8E 00 :6E
7460 03 7F E1 FF FF FF FF FB FF F3 FF E1 FF C0 7F F8 00 :63
7470 00 00 00 00 00 00 00 00 00 20 00 C0 01 80 03 00 :64
7480 00 00 00 00 00 0F E0 03 C0 00 00 18 00 00 18 :E2
7490 00 00 F0 06 00 08 00 00 00 00 00 00 00 00 00 00 :FE
74A0 00 00 00 00 00 00 00 00 00 00 00 18 00 F0 06 :0E
74B0 00 08 00 00 00 01 F0 03 F8 07 F1 FF C3 FF 87 FE :32
74C0 0F F0 1F C0 3C 00 00 00 18 00 F0 06 00 08 00 :30
74D0 00 00 00 00 00 00 00 00 00 00 00 00 08 00 1C :24
74E0 00 00 00 00 00 FE 00 0C 00 78 07 A0 3F 40 7C 4E :72
74F0 03 FF 0F 38 06 03 E0 0C 60 0F 10 00 E0 00 C0 00 :5D
-----------------------------------------------------------
Sum: 42 E6 10 E7 09 FF 68 E0 6A A3 DE 7A 97 D8 81 84 :48

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7500 70 00 FE 00 0C 00 78 07 A0 3F 40 7C 4E 03 FF 0F :F3
7510 38 06 03 E0 0C 60 0F 10 00 E0 00 C0 00 70 00 FE :BA
7520 00 0C 00 18 00 20 00 40 00 00 00 00 00 00 00 00 :84
7530 00 00 00 00 00 00 00 00 00 00 00 8C FC 00 00 03 :8B
7540 00 07 E0 07 F8 13 8E 3C 3F 18 FE C4 78 1E 00 FE :70
7550 04 F8 0F 80 1F 00 38 00 00 FC 00 00 03 00 07 E0 :C8
7560 07 F8 13 8E 3C 3F 18 FE C4 78 1E 00 FE 04 F8 0F :94
7570 80 1F 00 38 00 00 FC 00 00 00 00 00 00 00 00 00 :D3
7580 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7590 00 00 00 00 40 01 C0 00 00 00 00 FE 03 8E 03 F8 :8B
```

キャラクタ＆音楽データ ダンプ・リスト

```
75A0 00 00 00 00 00 00 00 00 00 00 00 00 00 1F FC 00 :1B
75B0 00 40 01 C0 00 00 00 00 FE 03 FE 03 F8 00 00 07 :02
75C0 FE 01 FC 00 08 00 00 00 00 00 1F FC 00 00 00 00 :1E
75D0 0C 07 F8 FF F3 FF FF 8F FF FF FF FF 78 00 7E 00 :7C
75E0 1F F0 1E 30 3C 00 F8 1F FC 00 C0 04 00 1C 00 00 :8C
75F0 00 00 00 FE 03 8E 03 F8 00 00 00 00 00 00 00 00 :8A
---------------------------------------------------------
Sum: 5C 60 16 32 E5 60 1B 37 9C AD 38 8C 36 5E 7B FC :B3

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7600 00 00 00 00 00 01 FF 00 04 00 1C 00 00 00 00 00 :20
7610 FE 03 FE 03 F8 00 00 FF C1 FE 02 00 00 00 00 00 :BA
7620 00 00 01 FF 00 00 01 80 03 FC 07 FF 8F FF 9F 8F :42
7630 FF FF FF FF 00 3C 01 F1 FF 06 3C 00 78 00 F8 01 :DC
7640 FF 00 00 00 00 00 00 80 02 80 04 80 09 00 14 00 :A2
7650 10 00 00 00 00 00 00 00 00 00 00 00 00 FC 0C 00 :18
7660 00 00 00 00 80 03 80 07 80 0F 00 1C 00 10 00 02 :C7
7670 00 1C 00 18 00 10 00 00 00 00 FC 0C 00 0C 00 FC :54
7680 03 F8 0E F0 3C E0 79 C0 F7 81 FF 03 FC 03 E0 03 :AA
7690 E0 03 E0 03 60 06 40 FC 0C E3 00 00 00 00 40 01 :98
76A0 40 04 80 09 00 0A 00 08 00 00 00 00 00 00 00 00 :DF
76B0 00 00 00 00 07 E0 00 00 00 00 00 40 01 C0 07 80 :6F
76C0 0F 00 0E 00 08 00 00 01 80 02 00 00 00 00 00 00 :A8
76D0 00 07 E0 00 C0 01 F0 03 F0 07 70 0C F0 19 E0 3B :32
76E0 C0 7F 80 FF 00 7C 01 F0 07 C0 09 80 03 00 07 E0 :65
76F0 00 60 00 30 00 18 00 00 00 00 20 00 00 00 FE 00 :C6
---------------------------------------------------------
Sum: FE 03 DA 44 E3 B5 2B AF C3 BC F9 76 00 F3 C3 2D :62

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7700 00 00 00 00 00 00 30 00 C0 03 00 4C 01 E0 60 00 :80
7710 30 00 18 00 00 00 00 20 00 40 00 FE 00 00 00 00 :A6
7720 00 00 00 30 00 C0 03 00 4C 01 E0 00 02 00 04 00 :26
7730 08 00 30 00 60 00 80 03 80 07 00 06 00 0C 00 08 :BC
7740 00 10 00 20 00 40 01 E0 0C 00 0C 00 60 03 00 00 :CC
7750 08 00 00 00 20 00 00 3F 00 00 00 00 00 00 06 00 :6D
7760 06 10 06 30 06 00 00 0C 00 60 03 00 00 08 00 10 :D9
7770 00 20 00 00 3F 00 00 00 00 00 00 06 00 06 10 06 :81
7780 30 06 00 80 01 00 02 00 06 00 0C 00 08 00 38 00 :0B
7790 70 00 C0 01 80 02 00 04 00 08 00 10 00 30 00 00 :FF
77A0 00 00 00 00 00 00 00 F0 1F 83 1E 1C 20 00 00 00 :EC
77B0 00 00 00 00 07 FC 00 00 00 00 00 00 00 00 00 00 :03
77C0 00 00 00 00 F0 1F 83 FE 1F E0 00 00 03 FE 00 00 :90
77D0 00 07 FC 00 00 00 00 00 00 00 00 00 00 00 00 00 :03
77E0 00 00 3F E7 1F FC 3F FF FF 7C 00 1F FC F8 07 FC :10
77F0 00 00 00 00 00 00 00 C1 00 00 00 00 00 07 00 07 :CF
---------------------------------------------------------
Sum: E6 4D 49 E8 5C 19 78 00 DB 92 19 A1 8A 2A B9 21 :06
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7800 F8 07 C6 01 0F 00 00 00 00 00 00 00 00 1F F0 00 :E4
7810 00 00 00 00 00 00 00 00 00 00 07 00 07 F8 07 FE :0B
7820 01 FF 00 00 FF E0 00 00 00 00 1F F0 00 00 00 00 :EE
7830 00 00 00 00 00 00 00 00 00 0F F8 1F C7 3F 0F FF :3A
7840 FF 00 1D FF E0 03 E0 1F F0 00 00 00 00 00 00 60 :4D
7850 0E 60 1C 78 00 00 00 00 00 14 00 00 00 00 00 00 :16
7860 00 00 00 00 10 03 E0 04 00 F8 01 E0 60 0E 60 1C :BA
7870 78 00 00 00 00 00 1C 00 00 7F FF 00 00 00 00 00 :12
7880 00 10 03 E0 04 00 F8 01 E0 00 00 00 00 00 00 00 :D0
7890 00 07 E0 77 FF FF 80 00 07 C0 00 00 00 00 00 00 :A3
78A0 00 00 00 00 00 00 87 00 3E 01 C0 0E 00 00 00 00 :94
78B0 14 00 00 00 00 00 00 00 00 00 00 00 07 00 07 98 :BA
78C0 01 B0 01 80 00 3E 01 C0 0E 00 00 00 00 1C 00 00 :5B
78D0 00 7F FF 00 00 00 00 00 00 07 00 07 98 01 B0 01 :D6
78E0 80 00 00 00 00 00 00 00 0C 00 17 00 3F FF 80 00 :61
78F0 07 C0 00 00 00 00 00 00 00 00 00 00 00 00 87 17 :65
---------------------------------------------------------
Sum: 1A 6C E2 4F 01 23 DC E4 2F 62 F5 04 0C 80 24 29 :FE

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7900 02 E7 B7 02 74 59 02 E7 B7 02 74 59 02 E7 B7 02 :80
7910 74 59 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :CD
7920 01 74 00 01 99 00 01 B7 00 01 E7 00 00 00 00 00 :AF
7930 02 E7 B7 02 74 59 02 E7 B7 02 74 59 02 E7 B7 02 :80
7940 74 59 00 00 00 02 74 00 02 70 00 02 74 00 02 70 :9D
7950 00 02 74 00 02 70 00 02 74 00 00 00 00 00 00 00 :5E
7960 02 74 74 02 79 79 02 80 80 02 88 88 00 00 00 00 :F2
7970 02 E7 E7 02 E7 74 02 74 99 02 B7 99 02 E7 E7 02 :60
7980 E7 74 02 74 99 02 B7 99 00 00 00 00 00 00 00 00 :BC
7990 08 99 B7 06 B7 E7 02 AC CD 08 99 B7 08 E7 E7 02 :A7
79A0 CD F0 02 B7 E7 02 AC CD 02 99 B7 04 AC CD 04 B7 :62
79B0 E7 10 CD F0 02 B7 E7 02 AC CD 02 99 B7 02 AC 88 :57
79C0 04 99 B7 04 99 B7 08 74 99 08 99 B7 04 AC CD 04 :96
79D0 B7 E7 06 CD F0 02 E7 E7 10 E7 E7 00 00 00 00 00 :0F
79E0 06 B7 E7 02 AC CD 08 99 B7 06 CD F0 02 B7 E7 08 :E2
79F0 AC CD 04 B7 E7 04 B7 E7 02 CD F0 02 B7 E7 02 AC :CA
---------------------------------------------------------
Sum: 01 62 6D B4 38 3D 77 6A DA A9 9D D2 A2 B5 A4 6F :36

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7A00 CD 02 CD F0 08 E7 E7 00 00 00 00 00 00 00 00 00 :62
7A10 99 99 CD 66 79 CD 99 66 66 99 79 66 4C CD 59 66 :60
7A20 66 99 59 66 74 CD 66 66 79 99 74 66 88 CD 79 66 :EB
7A30 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :FF
---------------------------------------------------------
Sum: CB 34 F3 BC F5 81 E6 CC DF 32 ED CC D4 9A D2 CC :AC
```

RANDOM BOX

FM-7/11 Kコンパイラ用三角関数サブルーチン

■渡辺誠太郎

Kコンパイラ用にSIN，COS関数を作ってみました．

● **ステートメントの文法**

 ＊SIN〔角度〕

 ＊COS〔角度〕

リスト1 三角関数サブルーチン

```
1000 '(* K-TRAGONOMETRICAL FUNCTION SUB-ROUTINE
*)
1010 'COS;THETA=90-%1;GOTO ALPHA
1020 'SIN;THETA=%1
1030 'ALPHA;THETA=MOD[THETA,360];IF THETA<0 THEN
 THETA=360+THETA;FI
1040 'PSI=THETA;IF PSI>270 THEN PSI=360-PSI;GOTO
 BETA;FI
1050 'IF PSI>180 THEN PSI=PSI-180;GOTO BETA;FI
1060 'IF PSI>90 THEN PSI=180-PSI;FI
1070 'BETA;PHI=FUCN[PSI];IF THETA>180 THEN PHI=-
PHI;FI
1080 'DMY=PHI;RETURN
1090 'FUCN;CODE $30,$8D,$00,$05,$EC,$62,$E6,$8B,
$39
1100 'CODE 0,2,3,5,7,9,10,12,14,17,17,19,21,23,2
4,26,28,29,31,33,34,36,37,39,41,42,44,45,47,48,5
0,52,53,54,56,57,58,60,62,63,64,66,67,68,69,71,7
2,73,74,75,77,78,79,80,81,82,83,84,85,86,87,87,8
8,89,90,91,91,92,93,93
1110 'CODE 94,95,95,96,96,97,97,97,98,98,98,99,9
9,99,99,100,100,100,100,100,100
1120 'MOD;DMY=%1-%1/%2*%2;RETURN
```

機能：与えられた角度のSIN関数，COS関数を得る．F-BASICと異なり，角度にはラジアンではなく度を用いる．得られた関数は，本来の値０～１の100倍（＝０～100）になっている．したがって，メイン・プログラム側でリスケールすること．

例：A＝SIN〔60〕

　　B＝COS〔−750〕

参考文献

1）“拡張コンパイラ”FM-8活用研究，工学社

2）“Kコンパイラ用グラフィック・サブルーチン”FM-7/8活用研究，工学社

3）“正弦・余弦表”応用数字，日本書院

リスト2 デモ・プログラム

```
10 'CLS[0,7,0]
20 'FOR L=0 TO 5
30 '    FOR R=0 TO 360 STEP 2
40 '        Y=100-80*SIN[R+60*L]/100
50 '        PSET[R+100,Y,7]
60 '    NEXT
70 'NEXT
80 'END
90 'CLS;HLT[];CPOKE ADDR,$02,%1,%3,%2;SUB[];RETU
RN
100 'PSET;HLT[];CPOKE ADDR,$17,1,%1#,%2#,%3,0;SU
B[];RETURN
110 'HLT;CODE $B6,$FD05#,$2BFB#;POKE $FD05,$80;C
ODE $B6,$FD05#,$2AFB#;ADDR=$FC82;RETURN
120 'SUB;POKE $FD05,0;RETURN
```

FM-7

問題数60問！ ウカウカしてると制限時間

詰将棋道場

■土肥一夫

◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

詰将棋ゲームは以前，ベストセラーになったこともあるほどで，思考型ゲームの中では麻雀の次に人気のあるものです．しかし，市販されている詰将棋ソフトでは，問題数が少なすぎて，さっぱり楽しめなかった方が多いと思います．

今回，問題数，機能，美しさの点でも数段上まわる（と私は思う）ものを作りましたので，発表したいと思います．

◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

特徴

詰将棋ゲームは，すでに何本か市販されていますが，大別して，

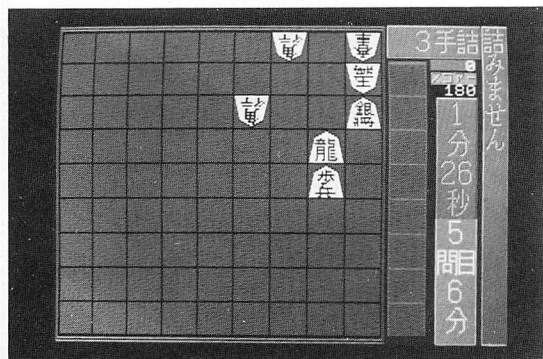
❶プログラムにあらかじめ問題が内蔵されていて，コンピュータが問題を出し，人がそれを解くもの．

❷乱数によって自動的に問題を作って，人がそれを解くもの．

❸人が問題を出してコンピュータが解いてくれるもの．

があります．❷❸のレベルはまだ高くなく，内容的には大変興味深いものの，まだ使い物にならないのではと思います．やはり楽しめるものといえば，❶のタイプのプログラムになります．

しかしながら，❶のタイプのものにも，現時点のレベルでは，2つの大きな欠点があります．

❹問題数が少なすぎて楽しめない（20問以下のものが多い）．

❺疑問手（つまり間違った指し手）に対して，1手分しか相手をしてくれないので，どこで間違えたか，すぐにわかってしまう．

これらの欠点は，詰手順のデータが充分に圧縮されていないためにメモリ容量不足になってしまうことと，プログラム面での工夫が足りないことが原因であるような気がします．

❻駒が動けないような位置を間違えて入力したり，王手になっていない位置にうっかり駒を動かした場合でも，あたかも本気で指したように扱われてしまう．

今回制作した詰将棋プログラムは，これらの欠点を克服するために．

❶プログラムで，王手チェックや動けるかどうかのチェックをする．

❷詰手順のデータとして，動かす駒の元の位置を省略することによってデータ量を減らす．

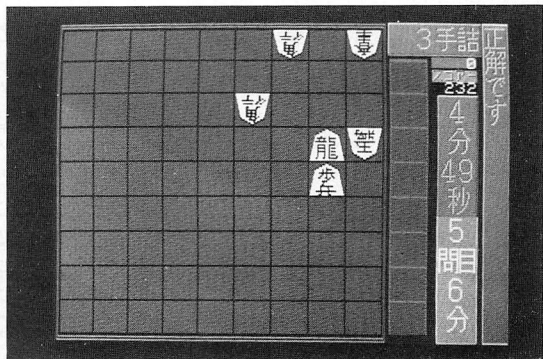
❸あまり見込みのない手は無駄な手として登録せず，データの爆発的増加を防ぐ

❹その代わり，疑問手でも2手以上相手になって応じる手を指すようにする．

このことにより問題数，なんと60問を実現しました(100問まで可能)．また表示は大変美しく，この点でも相当レベルのものです．メッセージも的確に出すようにし，『詰みません』,『正解です』のほかに『動かせません』,『成れません』,『最短手順ではありません』を場合に応じて，漢字かなまじりで表示します．

特に必要ではないかも知れませんが，バック・グラウンドで音楽を鳴らして，ゲームとしての楽しさを増すように

ウッ！この手もダメか

やった！これで詰まった

もしました.

もうひとつ私が狙ったのは，詰将棋ゲームのシステム化です．60問の詰将棋をアマチュアの私が作ることも骨の折れることですが，それ以上に，詰将棋をいかに能率よくデータに変換するかが問題でした．そこで，**データ作成コンパイラ**なるものを作って，データ入力の労力とミスを減らしました．

もしあなたが，詰将棋を作る力があれば，詰将棋コンパイラの助けを借りて，容易にあなたの『詰将棋道場』を作ることができるわけです．

後日，新しい問題と共に『詰将棋コンパイラ』を発表したいと思います．

プログラムの入力と起動

プログラムは3本構成になっています．まず，『**ショウギ**』のBASICプログラムと『**ショウギSB**』のマシン語プログラムを入力して，それぞれ，

```
SAVE"ショウギ"
SAVEM"ショウギSB", &H5500, &H7DFF, &H5F00
```

として1本のテープに入れます．

さらに『**ショウギDT**』を入力して，

```
SAVE"ショウギDT", &H6000, &H745F, &H6000
```

として問題テープを作ります．

次のことを注意してください．

①プログラムをテープからロードするとき，テレコのリモート端子をつなぐ．

②ロードする前に1度リセットする．

③RUN RETURN で起動する．

プログラムを途中で止めたり，終了したりしてから再び走らせるには，ただRUN CR とするだけで元どおり正常に動きます．テープの再セットは必要ありません．

遊び方

キー操作は8方向のカーソル方式で 1 ～ 9 キーで動かします．『.』を使って駒台と盤を1度に移動させることができます．

駒の動かし方

①まず動かしたい駒の位置に点滅するカーソルを動かします．

② CR キーを押します．

③動かす先にカーソルを動かします．

④成らないときには CR キー，成るときには 0 キーを押します．

他のコマンド

DEL …元の図に戻りたいときに押す．

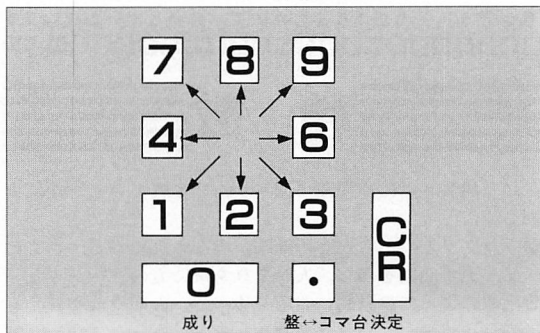
ESC …これ以上解ける問題がないときに押す．

CLS …1問前の図に戻りたいときに押す．

HOME …1問後へスキップしたいときに押す．

なお，起動時にたずねてくる．実力と問題数に CR で答えたときはそれぞれ4級，5問に設定されます．

級が高いとやさしい問題が主体になりますので，ときどき級を変えてやった方が面白味が増すと思います．

キー操作

リスト1　ショウギ

```
90 CLEAR 300,&H54FF:SCREEN7,7:WIDTH40,25
:CONSOLE0,25,0
100 RANDOMIZE VAL(MID$(TIME$,4,2))*60+VA
L(RIGHT$(TIME$,2))
120 DEFINTA-Z:DEFSNG S
160 IF PEEK(&H5F00)=&H20 THEN 300
200 LOADM "":EXEC
300 '
301 BEEP
302 INPUT "ベツ ノ テープ ノ モンダイニシマスカ。(Y/N)"
,K$
305 IF INSTR("Yyン",LEFT$(K$+" ",1))=0 TH
EN 400
310 PRINT:PRINT:PRINT"モンダイ ヲ ニュウリョク シマス
。"
320 PRINT:PRINT"モンダイ ノ テープ ヲ セット シテクダ
サイ。"
330 PRINT:PRINT"テープレコーダ ノ スイッチ ヲ イレテ、"
340 PRINT:PRINT"ドレデモ キー ヲ オシテクダサイ。"
350 K1$=INPUT$(1):GOTO 410
400 IF PEEK(&H5FFF) THEN 420
410 LOADM "":BEEP
420 POKE &H5FFF,1
500 INPUT "アナタ ノ ジツリョク ハ ナン キュウ デスカ(1-
9)",K$
505 IF K$="" THEN LVL=5 ELSE IF K$<"1" O
R K$>"9" THEN 500 ELSE LVL=ASC(K$)-&H30
510 PRINT:PRINT:INPUT"モンダイスウ ハ ナンモン ニ シ
マスカ。",K$
515 IF K$="" THEN NMX=4 ELSE IF LEN(K$)>
2 THEN GOTO 510 ELSE NMX=VAL(K$)-1
520 IF NMX<0 OR NMX>=100 THEN 510
910 DIM PTN(200)
990 N0=100
1000 DIM BD(9,8),BD0(9,8),DT(100),DR(15)
,PB(N0,3),CG(32)
1020 MRST=&H5520:DEF USR=MRST
1030 MUST=&H5600:DEF USR1=MUST:DEF USR2=
MUST+3:DEF USR3=MUST+6:DEFUSR4=MUST+9:A=
USR1(1)
1035 CGST=&H55A0:DEF USR5=CGST
1036 KDR=&H5E00:DEF USR7=KDR:DEFUSR8=KDR
+2:DEF USR9=KDR+4:KOMA=KDR+6
1040 DEF FNR(X)=INT(RND(1)*(X+1))
1050 DEF FNXPS(X)=INT((X+13) MOD 9)-4
1060 DEF FNYPS(X)=INT((X+13)¥9)-1
1070 DEF FNSS$(X)=RIGHT$(STR$(X+100),2)
1100 XST=10:YST=5:XSP=50:YSP=21
1110 MOX=XST+XSP*9+10:MOY=YST+YSP
1115 MEX=XST+XSP*9+10
1120 MESPX=604:CLMS=7:MESPYS=2:MESPYE=19
7
1130 TMX=MOX+XSP+16:TMYS=50:TMYE=197:TMS
P=18
1140 SCX=33:SCY=5
1200 '-
1201 COLOR7,1:CLS
1202 LINE (70,5)-(550,30),PSET,3,BF
1205 LINE (70,5)-(550,30),PSET,7,B
1210 Y=10:X=100:FOR K=&H34 TO &H38
1220 GOSUB 8300:X=X+1:GOSUB 8300:X=X+99
1230 NEXT
1235 LINE (300,35)-(550,59),PSET,2,BF
1240 Y=40:X=310:FOR K=&H3C TO &H3F
1250 GOSUB 8300:X=X+1:GOSUB 8300:X=X+63
1260 NEXT
1265 LOCATE5,10:PRINT "キー ソウサ"
1270 PRINT:PRINT"1-9 .... ウゴカス"
1275 PRINT" .  .... バン <-> コマダイ"
1280 PRINT:PRINT" CR .... カクニン "
1290 PRINT" 0  .... ナル "
1292 PRINT:PRINT"HOME.... ツギ ノ モンダイ"
1294 PRINT"CLS .... マエ ノ モンダイ"
1296 PRINT"DEL .... ヤリナオシ"
1298 PRINT"ESC .... モウ コレイジョウ トケナイ"
1300 LOCATE0,24:PRINT"オンリョウ チョウセツ(1-9) ハ
ジメルトキハ ソノホカノ キー";
1310 K$=INPUT$(1)
1320 K=ASC(K$)-&H30
1330 IF K>=1 AND K=<9 THEN POKE &H560C,9
-K:GOTO 1300
1700 GOSUB 3500
1705 COLOR7,0:WIDTH40,25:CONSOLE0,25,0
1710 BCL=1:LINE(0,0)-(639,199),PSET,2,BF
1800 GOSUB 19000
1900 PW=0:MW=NMX+1
1910 SCORE=0
1955 GOSUB 18000:GOSUB 4000
1960 LOCATESCX,SCY-2:PRINTRIGHT$("    "+
STR$(SCMX),4)
1970 LINE(SCX*16,(SCY-2)*8)-((SCX+4)*16-
1,(SCY-2)*8+6),OR,BCL,BF
2000 GOTO 2010
2005 '
2006 GOSUB 11600
2008 GOSUB 4000
2010 '
2011 K=PB(PW,3):TIM$="00:"+FNSS$(K¥60)+"
:"+FNSS$(K MOD 60):MCT=0
2012 GOSUB 3000
2013 GOSUB 11000
2015 GOSUB 12000
2020 MISC=2:TIME$=TIM$:A=USR1(0)
2021 GOTO 2025
2022 '- redo
2023 ADDR0=ADDR00:GOSUB 11500
2024 MISC=2
2025 '-
2026 EF=0
2027 GOSUB 8200
2030 GOSUB 10000
2035 IF AF THEN IF AB=-1 THEN 2170 ELSE
IF AB=1 THEN 2150 ELSE IF AB=0 THEN PW0=
PW:GOTO 2100 ELSE IF AB=2 THEN PW0=PW:MW
=0:GOTO 2180
2040 IF MISC<>0 THEN GOSUB 7000:IF OK=0
THEN GOSUB9000:GOSUB8100:GOTO 2100
```

リスト1　ショウギ

```
2042 IF OK AND NXF=0 AND MISC>1 THEN FA=
FAF:MISC=1
2048 IF MISC=0 THEN FOR I=0 TO 8000:NEXT
2050 GOSUB 9000
2052 IF MISC=0 THEN FAF=FA:GOSUB 8000:GO
TO 2100
2060 GOSUB 6000
2070 IF EF THEN 2120
2075 IF MISC=1 THEN MISC=MISC-1
2080 GOTO 2025
2100 ADDR=ADDR0
2110 GOTO2022
2120 ' tsumi
2122 PW0=PW
2125 K=USR1(2)
2127 FAF=5:GOSUB 8100
2130 PB(PW,2)=-1
2140 MW=MW-1:IF MW=0 THEN 2180
2150 '-step
2152 IF FL THEN PB(PW,2)=-1:MW=MW-1:IF M
W=0 THEN 2260
2155 PW0=PW
2160 PW=PW+1:IF PW>NMX THEN 2175 ELSE IF
 PB(PW,2)<>0 THEN 2160
2165 GOTO 2180
2170 '
2171 PW0=PW
2175 PW=PW-1:IF PW<0 THEN 2160 ELSE IF P
B(PW,2)<>0 THEN 2175
2180 ' step
2190 PB(PW0,3)=TIME:IF PB(PW0,3)>=3600 T
HEN PB(PW0,3)=0
2200 IF EF THEN GOSUB 3700
2250 IF MW>0 THEN 2005
2260 IF SCORE>SCMX THENSCMX=SCORE
2270 LOCATE10,23:PRINT"ｵﾜﾘﾆ ｼﾏｽｶ(Y/N)":K
$=INPUT$(1)
2280 IF INSTR("Yyﾝ",K$+" ")=0 THEN 1200
2290 A=USR2(A)
2300 END
3000 '-
3010 LINE (MEX,3)-(MEX+63,18),PSET,1,BF
3012 K=LE ¥ 10:IF K=0 THEN K=&H2A ELSE K
=32+K
3014 X=MEX:Y=3:GOSUB 8250
3015 K=(LE MOD 10)+32
3016 X=MEX+32:Y=3:GOSUB 8250
3018 K=LV*(3+LVL)¥4:TMB=K
3020 TMS$=STR$(K):TMS$=RIGHT$(TMS$,LEN(T
MS$)-1):IF LEN(TMS$)=1 THEN TMS$="0"+TMS
$
3021 TMS$="00:"+TMS$+":00"
3023 X=TMX:Y=TMYS+TMSP*4:K2$=RIGHT$(" "+
STR$(PW+1),2):GOSUB 10950
3024 X=TMX:Y=TMYS+TMSP*6:K2$=MID$(TMS$,4
,2):GOSUB 10950
3025 FOR I=0 TO 1
3026 X=TMX:Y=TMYS+TMSP*I*2:K2$=MID$(TIM$
,I*3+4,2):GOSUB 10950
3028 NEXT
3030 LINE (TMX-1,TMYS+TMSP*4)-(TMX+57,TM
YE+1),OR,3,BF
3040 RETURN
3500 '-
3510 M=0:K=&H6004:K1=1
3530 '
3540 PB(M,0)=K
3545 PB(M,1)=PEEK(K-2)*&H100+PEEK(K-1):P
B(M,2)=0:PB(M,3)=0
3548 K1=PEEK(K-4)*&H100+PEEK(K-3):K=K+K1
3550 M=M+1:IF K1<>0 THEN 3530
3600 NBD=M-2:IF NMX>NBD THEN NMX=NBD
3602 NMX1=NMX*2:IF NMX1>NBD THEN NMX1=NB
D
3610 FOR I=0 TO NMX1
3615 J=FNR(NBD)
3620 SWAP PB(I,0),PB(J,0)
3621 SWAP PB(I,1),PB(J,1)
3630 NEXT
3647 '
3648 IF NMX1=0 THEN 3682
3650 FOR J=0 TO NMX1-1
3655 FOR I=J+1 TO NMX1
3660 IF PB(J,1)>PB(I,1)+&H100 THEN SWAP
PB(I,0),PB(J,0):SWAP PB(I,1),PB(J,1)
3670 NEXT
3680 NEXT
3682 NMS=(NMX*(LVL-1)+NMX1*(9-LVL))¥8-NM
X
3684 FOR I=0 TO NMX:J=NMS+I
3686 PB(I,0)=PB(J,0)
3687 PB(I,1)=PB(J,1)
3688 NEXT
3690 RETURN
3700 '-- ｾｲｾｷ
3707 TM=VAL(MID$(TIME$,4,2))
3710 N1=SCORE:N1$=STR$(N1)
3715 SC1=(LV+11-LVL)*(110¥(NMX+11))*2
3720 SCORE=SCORE+SC1-(TM<TMB)*(SC1*(((TM
B-TM)*10)¥TMB))¥10:N2$=STR$(SCORE):SCORE
=SCORE
3725 N3=N1
3730 N3=N3+2:LOCATESCX,SCY:PRINT RIGHT$(
"    "+STR$(N3),4)
3740 IF N3<SCORE THEN GOTO 3730
3790 RETURN
4000 '--
4005 ADDR00=PB(PW,0)
4030 K=PB(PW,1):LV=K¥&H100:LE=K MOD &H10
0
4040 RETURN
6000 '-
6005 IF EF THEN RETURN
6008 IF XS2=9 THEN KK=-NN2:GOTO6020
6010 KK=-BD(XS2,YS2):BD(XS2,YS2)=0:PSX=X
S2:PSY=YS2:NN=0:MF=0:GOSUB 17000
6020 IF KK<=8 AND FF2 MOD 2=1 THENKK=KK+
6
6025 IF KK=1 THEN XOH=XD2:YOH=YD2
6030 BD(XD2,YD2)=-KK:PSX=XD2:PSY=YD2:NN=
-KK:MF=0:GOSUB 17000:GOSUB 10850
6100 RETURN
7000 '-
7010 OK=0:NXF=0:FAF=0
7015 NUM=PEEK(ADDR0)
7020 ADDR=ADDR0+1
7030 F1=-1:GOSUB 7800:IF N<>BD(XS,YS) TH
EN F1=0 ELSE NN1=N:FF1=F
7040 GOSUB 7900:IFX<>XD OR Y<>YD THEN F1
=0 ELSE XD1=X:YD1=Y
7050 IF FF1 AND 2 THEN XS1=9:YS1=1:GOTO7
060
7052 IF (FF1 AND 8)=0 THEN IF F1 THEN GO
SUB 7300:GOTO 7054 ELSE 7062
7053 GOTO 7055
7054 IFF=0 AND( FF1 AND 2)=0 THEN XS1=9:
 YS1=1:GOTO 7060 ELSE 7060
7055 GOSUB 7900:XS1=X:YS1=Y
7060 IF F1=0 THEN 7062
7061 IF((XS=XS1 AND YS=YS1) OR(XS=9 AND
XS1=9)) AND NF=(FF1 MOD 2=1) THEN 7100
7062 FF2=PEEK(ADDR):D=2
7063 IF FF2=0 THEN D=1:GOTO7068
7064 IF FF2 AND &H20 THEN D=2:GOTO 7068
7065 IF (FF2 AND &H40) THEN D=1
7066 IF(FF2 AND &H80) THEN D=D+1
7068 ADDR=ADDR+D+1
7070 N=PEEK(ADDR):IF N=0 THEN RETURN
7080 GOTO 7030
7100 '-OK
7110 OK=-1
7130 GOSUB 7800:NN2=-N:FF2=F:IFN=0 AND F
=0 THEN EF=-1:RETURN
7132 IF FF2 AND 2 THEN GOSUB 7900:XD2=X:
YD2=Y:XS2=9:YS2=8:BD(XS2,YS2)=NN2:GOTO 7
150
7134 IF FF2 AND 4 THEN XD2=XD1:YD2=YD1 E
LSE GOSUB 7900:XD2=X:YD2=Y
7136 IF( FF2 AND 8)=0 THEN GOSUB 7350:IF
 F=0 THEN XS2=9:YS2=8:BD(XS2,YS2)=NN2 EL
SE A=A ELSE GOSUB 7900:XS2=X:YS2=Y
7150 FAF=PEEK(ADDR):IF FAF>=&H80 THEN FA
F=FAF-&H80:NXF=0:RETURNELSE NXF=-1:LN=FA
F
7160 ADDR0=ADDR0+NUM
7170 LN=LN-1:IF LN>0 THEN NUM=PEEK(ADDR0
):GOTO 7160
7200 RETURN
7300 '-
7310 XB=XD1:YB=YD1:KS=NN1:D=-1:GOSUB 740
0
7320 XS1=XA:YS1=YA
7330 RETURN
7350 '-
7360 XB=XD2:YB=YD2:KS=NN2:D=1:GOSUB 7400
7370 XS2=XA:YS2=YA
7380 RETURN
7400 '-
7405 F=0
7410 POKE KOMA,ABS(KS):N1=USR9(0)
7415 FOR I=1 TO N1
7420 K=USR9(I):XA=XB-D*USR7(I):YA=YB-D*U
SR8(I):IF XA>=0 AND XA<=8 AND YA>=0 AND
YA<=8 THEN IF BD(XA,YA)=KS THEN I=N1:F=-
1:I=N1
7430 NEXT
7435 IF F THEN RETURN
7440 N2=USR9(N1+1):IF N2=0 THEN 7475
7445 FOR I=N1+2 TO N1+1+N2
7450 X=XB:Y=YB
7455 X=X-D*USR7(I):Y=Y-D*USR8(I)
7460 IF X<0 OR X>8 OR Y<0 OR Y>8 THEN 74
70
7465 IF BD(X,Y)=KS THENXA=X:YA=Y:F=-1:I=
N1+1+N2
7467 IF BD(X,Y)=0 THEN 7455
7470 NEXT
7475 RETURN
7800 '- N
7830 XY=PEEK(ADDR):ADDR=ADDR+1
7840 N=XY MOD16:F=XY ¥ 16
7850 RETURN
7900 '- X,Y
7930 XY=PEEK(ADDR):ADDR=ADDR+1
7940 X=(XY ¥16)-1 :Y=(XY MOD 16)-1
7950 RETURN
8000 '-
8020 GOSUB 8200
8025 X=MESPX:Y=MESPYS
8030 FOR K=MS(FAF,0) TO MS(FAF,1)
8034 GOSUB 8300
8040 Y=Y+16
8050 NEXT
8065 IF MS(FAF,2)<0 THEN 8090
8070 FOR K=MS(FAF,2) TO MS(FAF,3)
8074 GOSUB 8300
8080 Y=Y+16
8085 NEXT
8090 RETURN
8100 '-
8110 GOTO 8020
8200 '-
8210 LINE (MESPX,MESPYS)-(MESPX+31,MESPY
E),PSET,1,BF
8220 RETURN
8250 '
8260 LINE (X,Y)-(X+31,Y+15),PSET,BCL,BF
8270 GOSUB 8300
8280 RETURN
8300 '-
8310 POKE CGST+2,K:C=VARPTR(CG(0)):C=USR
5(C)
8320 PUT@ (X,Y)-(X+31,Y+15),CG,PSET,6
8330 RETURN
9000 '-MOVE
9010 '
9020 IF XS=9 THEN 9200
9030 KK=BD(XD,YD)
9040 IF KK=0 THEN 9100
9045 IF KK<-8THEN KK=KK+6
9050 BD(9,ME)=-KK:ME=ME+1
9060 MF=-1:PSX=XS:PSY=ME-1:NN=-KK:GOSUB
17000:MF=-1
9100 '
9110 MF=0:KK=BD(XS,YS):BD(XS,YS)=0:PSX=X
S:PSY=YS:NN=0:GOSUB 17000
9120 MF=0:PSX=XD:PSY=YD:KK=KK-NF*6:BD(XD
,YD)=KK:NN=KK:GOSUB 17000:GOSUB 10850
9190 RETURN
9200 '-
9202 KK=BD(XS,YS)
9203 FOR I=0 TO ME-1
9204 IF KK=BD(9,I) THEN YS=I:I=ME-1
9206 NEXT
9210 FOR I=YS TO ME-2
9220 BD(9,I)=BD(9,I+1)
9230 NEXT
9240 BD(9,ME-1)=0
9245 MF=-1
9250 FOR I=YS TO ME-1
9260 PSX=9:PSY=I:NN=BD(9,I):GOSUB 17000
9270 NEXT
9275 MF=0:ME=ME-1
9290 GOTO 9120
10000 '-
10002 IF INKEY$<>"" THEN 10002
10005 PSXS=9:PSYS=0:SF=0:FL=0
10007 PSX=XOH:PSY=YOH
10010 FF=0:GOSUB 10500:IF AF<>0 THEN 102
20
10015 IF BD(PSX,PSY)<=0 THEN 10010
10020 RCL=2:GOSUB 10800
10030 KS=BD(PSX,PSY):XS=PSX:YS=PSY
10040 FF=-1:GOSUB 10500:IF AF<>0 THEN102
00
10050 XD=PSX:YD=PSY
10060 GOSUB 15000
10062 IF NF AND((YS>2 AND YD>2)OR XS=9 O
R BD(XS,YS)>8) THEN FAF=4:GOSUB 8100:GOS
UB 10200:GOTO 10000
10070 IF OK=0 THEN FAF=3:GOSUB 8100:GOSU
B10200:GOTO10000
10080 GOSUB 16000
10085 GOSUB 10200
10090 IF OK=0 THEN FAF=2:GOSUB 8100:GOTO
10000
10100 RETURN
10200 '-
10210 PSX=XS:PSY=YS:RCL=2:GOSUB 10800
10220 RETURN
10500 '-
10510 AF=0:AB=0:NF=0
10520 I=10:J=1:RCL=6:GOSUB 10800
10530 K$=INKEY$
10535 IFI=1 THEN IFTIM$<>TIME$ THEN TIMS
$=TIM$:TIM$=TIME$:GOSUB 10900:IF F THEN
K$=CHR$(11):FL=-1
10540 IF K$="" THEN I=I-1:IF I<0 THEN I=
20:J=1-J:RCL=6:GOSUB 10800:GOTO 10530 EL
SE 10530
10550 IF J THEN RCL=6:GOSUB 10800
10552 GOSUB 10860
10553 PSX1=PSX:PSY1=PSY
10554 IF K$=CHR$(27) THEN AF=-1:AB=2:GOT
O10640
10555 IF K$=CHR$(127) THEN AF=-1:AB=0:GO
TO10640
10556 IF K$=CHR$(11) THEN AF=-1:AB=1:GOT
O10640
10557 IF K$=CHR$(12) THEN AF=-1:AB=-1:GO
TO10640
10558 IF K$=CHR$(13) THEN GOTO10640
10559 IF K$="0" AND FF THEN NF=-1:GOTO 1
0640
10560 IF K$="." THEN SWAP PSX1,PSXS:SWAP
 PSY1,PSYS:SF=1-SF
10565 IF INSTR("123",K$) THEN PSY1=PSY1+
1
10570 IF INSTR("789",K$) THEN PSY1=PSY1-
1
10580 IF INSTR("147",K$) THEN PSX1=PSX1-
1:IF PSX1=8 THEN PSY1=PSY1+1:IF SF THEN
SF=1-SF:PSXS=9:PSYS=0
10590 IF INSTR("369",K$) THEN PSX1=PSX1+
1:IF PSX1=9 THEN PSY1=PSY1-1
10600 IF PSX1=9 AND PSY1>=0 AND PSY1<8 T
HEN 10620
10610 IF PSX1<0 OR PSX1>8 OR PSY1<0 OR P
SY1>8 THEN 10520
10620 '--
10630 PSX=PSX1:PSY=PSY1:GOTO 10520
10640 '--
10650 RETURN
```

リスト1　ショウギ

```
10800 '-
10805 IF PSX=9 THEN MF=-1 ELSE MF=0
10810 GOSUB 17800
10820 LINE(X+2,Y+1)-(X+XSP-1,Y+YSP-1),XO
R,RCL,BF
10830 RETURN
10850 '
10852 IF MCT<30 THEN A=USR3(A)
10854 SOUND 7,&HFF:SOUND 0,24:SOUND 1,0:
SOUND 2,32:SOUND 3,0:SOUND 4,26:SOUND 5,
0:SOUND 6,2:SOUND 8,16:SOUND 9,16:SOUND
10,16:SOUND 13,0:SOUND 11,0:SOUND 12,10:
SOUND 7,&HF0:FOR I1=0 TO 50:NEXT:SOUND 7
,&HFF:SOUND 7,&H80:FOR I1=0 TO 300:NEXT:
SOUND 7,&HFF
10856 IF MCT<30 THEN SOUND 7,&HF8:A=USR4
(A)
10858 RETURN
10860 '
10861 IF MCT<30 THEN A=USR3(A)
10864 SOUND7,&HFF:SOUND 0,25:SOUND 1,0:S
OUND 8,16:SOUND 9,16:SOUND 10,16:SOUND 1
3,0:SOUND 11,0:SOUND 12,3:SOUND 7,&HFE:F
OR I1=0 TO 40:NEXT:SOUND 7,&HFF
10867 IF MCT<30 THEN SOUND 7,&HF8:A=USR4
(A)
10868 RETURN
10900 '-
10910 K1$=TIM$:F=0
10920 IF MID$(K1$,4,2)<>MID$(TIM$,4,2)
THEN  X=TMX:Y=TMYS:K2$=MID$(K1$,4,2):GOS
UB 10950:IF K1$>=TMS$ THEN F=-1
10930 X=TMX:Y=TMYS+TMSP*2:K2$=MID$(K1$,7
,2):GOSUB 10950:IF MCT<30 THEN MCT=MCT+1
 ELSE A=USR2(A)
10940 RETURN
10950 '-
10955 K=ASC(MID$(K2$,1,1))-&H10:IF K=32
OR K=16 THEN K=&H2A:GOSUB 8250:X=X+24:GO
SUB 8250:X=X-12:GOTO 10965 ELSE GOSUB825
0:X=X+24
10965 K=ASC(MID$(K2$,2,1))-&H10
10970 GOSUB 8250
10980 RETURN
11000 '-
11010 FOR X=0 TO 9:FOR Y=0 TO 8
11015 BD(X,Y)=0:BD0(X,Y)=0
11020 NEXT:NEXT
11040 ADDR=ADDR00
11050 N0=PEEK(ADDR):ADDR=ADDR+1:DT(0)=N0
11052 ADDR0=ADDR00+N0
11055 ME=0
11060 FOR I=1 TO (N0¥2)*3-1 STEP 3
11070 PSX=PEEK(ADDR)¥16:PSY=PEEK(ADDR)MO
D16:NN=PEEK(ADDR+1):ADDR=ADDR+2:IF NN>=1
28 THEN NN=NN-256
11071 DT(I)=PSX-1:DT(I+1)=PSY-1:DT(I+2)=
NN
11072 IF NN=-1 THEN XOH=PSX-1:YOH=PSY-1
11074 IF PSX=10 THEN ME=ME+1
11075 BD(PSX-1,PSY-1)=NN
11076 BD0(PSX-1,PSY-1)=NN
11080 NEXT
11085 ADDR00=ADDR0:XOH0=XOH:YOH0=YOH:ME0
=ME
11090 RETURN
11100 ' DATA
11105 DATA 3,5,2,-1,5,4,2,5,5,7,10,1,2
11110 DATA 1,5,4,2,4,2,2,10,1,2,10,2,2
11500 '-
11510 FOR I=0 TO 9:IF I=9 THEN MF=-1 ELS
E MF=0
11515 FOR J=0 TO 8
11520 KK=BD0(I,J):IFKK<>BD(I,J) THEN IF
(I=9 AND J=8)=0 THEN NN=KK:PSX=I:PSY=J:G
OSUB 17000:BD(I,J)=KK
11530 NEXT:NEXT
11540 XOH=XOH0:YOH=YOH0:ME=ME0
11550 RETURN
11600 '-
11610 FOR I=0 TO 9:IF I=9 THEN MF=-1 ELS
E MF=0
11615 FOR J=0 TO 8
11620 IF BD(I,J)<>0 THEN IF (I=9 AND J=8
)=0 THEN NN=0:PSX=I:PSY=J:GOSUB 17000
11630 NEXT:NEXT
11640 XOH=XOH0:YOH=YOH0:ME=ME0
11650 RETURN
12000 '-
12050 N0=DT(PS)
12060 FOR I=PS+1 TO PS+((N0-2)¥2)*3+1 ST
EP 3
12070 PSX=DT(I):PSY=DT(I+1):NN=DT(I+2)
12075 MF=0:IF PSX=9 THEN MF=-1
12080 GOSUB 17000
12090 NEXT
12100 RETURN
15000 '-
15001 OK=0
15003 K1=BD(XS,YS):IF YD=0 AND (K1=5 OR
K1=6) AND NF=0 THEN RETURN
15004 IF YD<=1 AND K1=4 AND NF=0 THEN RE
TURN
15005 K2=BD(XD,YD):IF XS=9 THEN IF K2=0
THEN OK=-1:RETURN ELSE RETURN
15007 IF K1>8 AND NF THEN RETURN
15010 XX=XS-XD:YY=YS-YD
15020 IF XX=0 AND YY=0 THEN 15200
15030 IF K2>0 THEN 15200
15035 XY=XX+YY*9
15037 POKE KOMA,KS:N1=USR9(0)
15050 FOR I=1 TO N1
15060 K=USR9(I):IF K=XY THEN OK=-1
15070 NEXT
15080 IF OK THEN GOTO 15200
15090 N2=USR9(N1+1):IF N2=0 THEN 15200
15100 FOR I=N1+2 TO N1+1+N2
15110 X=XS:Y=YS
15120 X1=-USR7(I)
15130 Y1=-USR8(I)
15140 X=X+X1:Y=Y+Y1
15150 IF X<0 OR X>8 OR Y<0 OR Y>8 THEN F
=-1:GOTO 15190
15160 K=BD(X,Y)
15170 IF X=XD AND Y=YD THEN OK=-1:I=N1+1
+N2:GOTO 15190
15180 IF K=0 THEN GOTO 15140
15190 NEXT
15200 '
15210 RETURN
16000 '-
16002 IF XS=9 THEN IF BD(XD,YD)<>0 THEN
OK=0:RETURN
16005 OK=0
16007 IF KS<=8 AND NF THEN KS=KS+6
16010 XX=XD-XOH:YY=YD-YOH
16020 IF XX=0 AND YY=0 THEN 16200
16035 XY=XX+YY*9
16037 POKE KOMA,KS:N1=USR9(0)
16050 FOR I=1 TO N1
16060 K=USR9(I):IF K=XY THEN OK=-1
16070 NEXT
16080 IF OK THEN GOTO 16200
16090 N2=USR9(N1+1):IF N2=0 THEN 16200
16100 FOR I=N1+2 TO N1+1+N2
16110 X=XD:Y=YD
16120 X1=-USR7(I)
16130 Y1=-USR8(I)
16140 X=X+X1:Y=Y+Y1
16150 IF X<0 OR X>8 OR Y<0 OR Y>8 THEN F
=-1:GOTO 16190
16160 K=BD(X,Y)
16170 IF X=XOH AND Y=YOH THEN OK=-1:GOTO
 16190
16180 IF K=0 THEN GOTO 16140
16190 NEXT
16200 '
16201 IF OK THEN RETURN
16202 IF XS=9 THEN RETURN
16250 BD1=BD(XS,YS):BD2=BD(XD,YD)
16260 BD(XD,YD)=BD1:BD(XS,YS)=0
16300 '-
16310 FOR J=0 TO 7
16320 DX=DR(J*2):DY=DR(J*2+1)
16325 D=DX+DY*9
16327 X=XOH:Y=YOH
16330 FOR N=1 TO 8
16340 X=X+DX:Y=Y+DY:IF X<0 OR X>8 OR Y<0
 OR Y>8 THEN N=8:GOTO 16400
16350 IF BD(X,Y)=0 THEN 16400
16355 IF BD(X,Y)<0 THEN N=8:GOTO 16400
16360 GOSUB 16500:N=8
16400 NEXT
16410 NEXT
16420 BD(XS,YS)=BD1:BD(XD,YD)=BD2
16450 RETURN
16500 '--
16520 K=BD(X,Y)
16530 POKE KOMA,K:N1=USR9(0):N2=USR9(N1+
1):IFN2=0THEN RETURN
16540 FOR I=N1+2 TO N1+N2+1
16550 K=USR9(I):IF K=D THEN OK=-1
16560 NEXT
16570 RETURN
17000 '-
17010 '
17020 GOSUB 17800
17025 N1=NN:IF N1<0 THEN N1=&H100+(N1 MO
D &H100)
17026 IF N1=0 THEN N1=15
17027 K=VARPTR(PTN(0))
17030 POKE MRST+2,N1:POKE MRST+3,K¥&H100
:POKE MRST+4,K MOD &H100
17037 K=USR(K)
17038 PUT@A(X+2,Y+1)-(X+XSP-1,Y+YSP-1),P
TN,PSET
17040 RETURN
17800 '-
17810 IF MF THEN X=MOX:Y=MOY+PSY*YSP:RET
URN
17820 X=XST+PSX*XSP:Y=YST+PSY*YSP:RETURN
18000 '-
18006 PSX=0:PSY=0:NN=15:GOSUB 17000
18008 FOR J=YST+1 TO YST+YSP*8+1 STEP YS
P:FOR I=XST+2 TO XST+XSP*8+2 STEP XSP
18009 PUT@A(I,J)-(I+XSP-3,J+YSP-2),PTN,P
SET
18010 NEXT:NEXT
18011 LINE(MOX,YST+YSP-1)-(MOX+XSP+1,YST
+YSP*9-1),PSET,4,BF
18012 FOR J=MOY+1 TO MOY+YSP*7+1 STEP YS
P
18013 PUT@A(MOX+2,J)-(MOX+XSP-1,J+YSP-2)
,PTN,PSET
18014 NEXT
18015 FOR J=2 TO 4 STEP 2
18016 LINE(XST-J,Y-J)-(XST+XSP*9+J,YST+Y
SP*9+J),PSET,6,B
18017 LINE(XST-J+1,Y-J+1)-(XST+XSP*9+J-1
,YST+YSP*9+J-1),PSET,2,B
18018 NEXT
18025 FOR I=0 TO 9
18027 LINE (I*XSP+XST,YST)-(I*XSP+XST+1,
YST+YSP*9-1),PSET,0,B
18030 NEXT
18040 FOR I=0 TO 9
18050 LINE (XST,I*YSP+YST)-(9*XSP+XST-1,
YST+YSP*I),PSET,0
18060 NEXT
18070 LINE (MESPX-3,MESPYS-2)-(MESPX+34,
MESPYE+2),PSET,1,BF
18080 LINE (MESPX-3,MESPYS-2)-(MESPX+34,
MESPYE+2),PSET,5,B
18100 LINE (TMX-1,TMYS-2)-(TMX+56,TMYE+2
),PSET,1,BF
18110 LINE (TMX-1,TMYS-2)-(TMX+56,TMYE+2
),PSET,6,B
18112 LINE (MEX-2,0)-(MEX+128,22),PSET,1
,BF
18114 LINE (MEX-2,0)-(MEX+128,22),PSET,6
,B
18120 FOR J1=0 TO 1:X=TMX+15:Y=TMYS+TMSP
+TMSP*J1*2:K=&H2C+J1:GOSUB 8300:NEXT
18130 X=TMX:Y=TMYS+TMSP+TMSP*4:K=&H32:GO
SUB 8300:X=X+28:K=&H33:GOSUB 8300
18140 X=TMX+15:Y=TMYS+TMSP+TMSP*6:K=&H2C
:GOSUB 8300
18174 X=MEX+32*2:Y=3:K=0:GOSUB 8300
18176 X=MEX+32*3:Y=3:K=1:GOSUB 8300
18180 LOCATESCX,SCY-1:PRINT"ｽｺｱ-"
18190 LINE(SCX*16,(SCY-1)*8)-((SCX+4)*16
-1,(SCY-1)*8+7),XOR,6,BF
18200 RETURN
19000 '-
19001 RESTORE 19500
19002 FOR I=0 TO 7
19003 READ DR(I*2),DR(I*2+1)
19004 NEXT
19110 RESTORE 20200
19120 READ N0
19130 FOR I=0 TO N0-1
19140 FOR J=0 TO 3:READ MS(I,J):NEXT
19150 NEXT
19210 RETURN
19500 '--
19510 DATA -1,-1,-1,0,-1,1,0,-1,0,1,1,-1
,1,0,1,1
20000 '-
20200 '
20210 DATA 6,6,10,-1,-1,11,18,8,10,19,24
,8,10,25,27,8,10,28,29,8,10,2,5,-1,-1
```

リスト2　ショウギSB

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5500 55 55 55 55 55 55 AA AA AA AA AA AA 55 55 55 55 :4E
5510 55 55 AA AA AA AA AA AA 00 00 00 00 00 00 00 00 :A6
5520 20 05 04 39 96 B8 00 34 01 1A 50 B7 FD 0F EE 8C :8C
5530 F2 A6 8C EE 2B 0A 10 8E 01 68 8D 49 8D 1B 20 14 :00
5540 40 8D 42 40 8D 13 30 C4 8D 20 30 88 78 8D 1B 30 :98
5550 88 78 8D 16 B6 FD 0F 35 81 34 76 C6 B4 A6 80 A7 :0C
5560 C0 A6 80 A7 C0 5A 26 F5 35 F6 34 36 C6 78 31 85 :4B
5570 31 3F 54 86 08 66 84 69 A4 4A 26 F9 66 80 31 3F :08
5580 5A 26 F0 35 B6 34 06 C6 78 4A 3D AE 8C 97 30 8B :E6
5590 30 8B 30 8B 35 86 00 00 00 00 00 00 00 00 00 00 :31
55A0 20 03 14 B0 00 34 01 1A 50 B7 FD 0F E6 8C F3 10 :BE
55B0 AE 02 AE 8C EE 4F 58 49 58 49 58 49 58 49 58 49 :4C
55C0 30 8B C6 20 34 04 A6 80 C6 08 48 34 01 69 21 69 :3D
55D0 A4 35 01 69 21 69 A4 5A 26 F0 31 22 6A E4 26 E6 :8E
55E0 32 61 B6 FD 0F 35 81 F5 35 F6 34 36 C6 78 31 85 :89
55F0 31 3F 54 86 08 66 84 69 A4 4A 26 F9 66 80 31 3F :08
---------------------------------------------------------
Sum: 04 55 E5 B1 10 D6 FB CE 78 42 EC B2 98 5B 84 87 :F4
```

リスト2　ショウギSB

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5600 16 00 90 16 00 58 16 00 65 16 00 72 05 5A 63 5A :33
5610 9D 5A 9D 5A D3 5A D3 5B 1B 5A 63 5A 9D 5A 84 0B :01
5620 03 04 0C 00 08 00 00 00 00 00 00 00 59 5A 9D 5A :C5
5630 D3 5A BC 0B 03 04 0C 02 09 00 00 00 00 00 00 00 :12
5640 5A 5A D3 5B 1B 5A FA 0E 01 08 18 04 0A 00 00 00 :8E
5650 00 00 00 00 00 00 46 00 6D 14 03 03 00 CC CC 07 :6C
5660 FF 17 01 8A 1A 10 30 8D 00 25 BF FF F8 39 CC 07 :6F
5670 FF 17 01 7A 1A 10 30 8D 00 15 BF FF F8 39 CC 07 :4F
5680 F8 17 01 6A 1C EF 31 8D 00 80 10 BF FF F8 39 B6 :78
5690 FD 03 3B 34 16 81 02 26 6F A6 03 A1 8D 01 95 24 :2E
56A0 67 30 8D 01 90 C6 0C 3D 30 8B C6 06 31 8D FF 5D :65
56B0 EE 81 EF A1 5A 26 F9 31 8D FF 52 33 8D FF 5A C6 :66
56C0 03 AE A4 AF C4 AF 44 AE 22 AF 42 31 24 6F 46 6F :F5
56D0 47 34 04 86 0B A0 E4 A7 4B 68 E4 86 06 A0 E0 A7 :85
56E0 4A 33 C8 14 5A 26 DA 86 03 A7 C9 56 5A FC FF F8 :4F
56F0 ED C9 56 5C 30 8D 00 12 BF FF F8 86 04 B7 FD 02 :2D
------------------------------------------------------------
Sum: AC E9 48 BF A2 8E CF 93 52 33 0E FD C7 93 31 E1 :2A
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8. +9 +A +B +C +D +E +F Sum
5700 CC 07 F8 17 00 E8 1C EF 35 96 A6 8D FF 4C A7 8D :52
5710 FF 49 86 03 A7 8D FF 42 33 8D FE FD 8D 17 33 8D :65
5720 FF 0B 8D 11 33 8D FF 19 8D 0B B6 FD 03 86 04 B7 :0F
5730 FD 02 1C EF 3B 34 36 6D 46 10 26 00 AD 6D 47 10 :09
5740 26 00 97 AE 44 AC 42 25 02 AE C4 EC 81 4D 26 10 :26
5750 6A 8D FF 06 6D 8D FF 03 10 26 00 90 EC 1F 30 01 :FA
5760 AF 44 A7 8D FE F0 1F 98 84 0F A0 8D FE 9E 2A 04 :56
5770 27 02 86 01 A7 47 54 54 54 54 E7 48 30 8D 00 9C :76
5780 A6 8D FE D2 8B 0A C6 00 5C 80 0C 2A FB 8B 0C E7 :E9
5790 8D FE C2 48 EC 86 44 56 6A 8D FE B9 26 F8 ED 8D :E7
57A0 FE B5 30 8D 00 66 A6 47 A6 86 E6 8D FE AB 3D 44 :8C
57B0 56 44 56 44 56 44 56 A6 48 3D 44 56 44 56 E7 49 :B3
57C0 A6 4A E6 8D FE 92 17 00 25 4C E6 8D FE 89 17 00 :8C
57D0 1D A6 4B E6 47 17 00 16 20 12 86 0A E6 47 5A E7 :98
57E0 47 17 00 0A A6 49 A7 46 20 02 6A 46 35 B6 34 06 :3B
57F0 B7 FD 0E 86 03 B7 FD 0D 86 00 B7 FD 0D F7 FD 0E :55
------------------------------------------------------------
Sum: 75 B8 6F 4A 26 89 C5 77 C4 A5 8C 78 60 EE 64 8E :7E
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5800 86 02 B7 FD 0D 86 00 B7 FD 0D 35 86 01 64 32 21 :03
5810 19 14 10 0E 0C 0B 0A 09 08 07 07 06 57 45 52 5F :DE
5820 4D C0 49 63 45 44 41 61 3D B6 3A 3F 36 FA 33 E4 :97
5830 30 FA 2E 3B 03 58 59 58 D9 58 D9 59 29 59 29 59 :06
5840 79 59 79 59 DB 59 DB 5A 1F 5A 1F 5A 63 5A 63 5A :79
5850 9D 5A 9D 5A D3 5A D3 5B 1B 00 4B 4F 48 4D 48 8F :6A
5860 00 49 4F 46 4D 46 8F 00 44 4F 46 4D 48 4F 49 4D :53
5870 00 4B 4F 4B 4D 4B 8F 00 4B 4F 48 4D 48 4F 48 4D :67
5880 00 49 4F 46 4D 46 4F 46 4D 00 44 4F 48 4D 4B 4F :15
5890 4B 4D 00 48 4F 48 4D 48 8F 00 46 4F 46 4D 46 4F :58
58A0 46 4D 00 46 4F 48 4D 49 8F 00 48 4F 48 4D 48 4F :58
58B0 00 48 4F 49 4D 4B 8F 00 4B 4F 48 4D 48 4F 48 4D :62
58C0 00 49 4F 46 4D 46 4F 46 4D 00 44 4F 48 4D 4B 4F :15
58D0 4B 4D 00 48 4F 48 4D 48 8F 00 3F 8D 3F 8D 00 3F :72
58E0 8D 3F 8D 00 3F 8D 3F 8D 00 3F 8D 3F 8D 00 3F 8D :55
58F0 3F 8D 00 3F 8D 3F 8D 00 3F 8D 3F 8D 00 3F 8D 3F :07
------------------------------------------------------------
Sum: DA A4 6C D7 49 EC 50 20 B5 35 B0 49 24 90 54 D4 :25
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5900 8D 00 3F 8D 3F 8D 00 3F 8D 3F 8D 00 3F 8D 3F 8D :55
5910 00 3F 8D 3F 8D 00 3F 8D 3F 8D 00 3F 8D 3F 8D 00 :C8
5920 3F 8D 3F 8D 00 3F 8D 3F 8D 00 3C 8D 3C 8D 00 3A :FC
5930 8D 3A 8D 00 3C 8D 3C 8D 00 3C 8D 3C 8D 00 50 8D :55
5940 3C 8D 00 3A 8D 3A 8D 00 3C 8D 3C 8D 00 3C 8D 3C :EE
5950 8D 00 37 8D 37 8D 00 37 8D 37 8D 00 38 8D 38 8D :27
5960 00 38 8D 38 8D 00 38 8F 38 8D 00 33 8F 33 8D 00 :98
5970 38 8D 38 8D 00 38 8D 38 8D 00 49 8F 49 4F 4B 4F :1E
5980 49 8D 49 4D 4B 4D 00 49 8F 49 4F 4B 4F 49 8D 4D :31
5990 8D 00 4B 8F 4B 4F 4D 4F 4B 8D 4B 4D 4D 4D 00 4B :F2
59A0 8F 4B 4F 4D 4F 50 4D 4E 4D 4D 4D 4B 4D 00 49 8F :07
59B0 49 4F 4B 4F 49 8D 49 4D 4B 4D 00 49 8F 49 4F 4B :F1
59C0 4F 49 8D 4D 8D 00 4B 8F 4B 4F 4D 4F 4B 8D 48 8D :BC
59D0 00 44 FF 43 4F 44 4F 46 4F 48 4F 00 3D 8F 38 8F :27
59E0 3D 8D 38 8D 00 3D 8F 38 8F 3D 8D 38 8D 00 3C 8F :7C
59F0 38 8F 3C 8D 38 8D 00 3C 8F 38 8D 3C 8D 38 8D 00 :73
------------------------------------------------------------
Sum: CC 28 C2 07 9B DF 66 12 11 D5 05 46 BF D7 27 89 :26
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5A00 3D 8F 38 8F 3D 8D 38 8D 00 3D 8F 38 8F 3D 8D 38 :B7
5A10 8D 00 3C 8F 38 8F 3C 8D 38 8D 00 3C 8F 38 8F 00 :3F
5A20 41 8F 38 8F 41 8D 38 8D 00 41 8F 38 8F 41 8D 38 :C7
5A30 8D 00 42 8F 38 8F 42 8D 38 8D 00 42 8F 38 8F 42 :93
5A40 8D 38 8D 00 41 8F 38 8F 41 8D 38 8D 00 41 8F 38 :84
5A50 8F 41 8D 38 8D 00 42 8F 38 8F 42 8D 38 8D 00 42 :90
5A60 8F 38 8F 00 4B EF 48 CF 49 4D 00 4B FF 44 FF 00 :CA
5A70 46 4F 48 4F 49 4D 4B 4D 49 8F 48 8D 00 46 FF 00 :4C
5A80 48 4F 49 4F 4B 4D 4D 4D 4B 8F 4B 8D 00 50 FF 4B :AD
5A90 FF 00 49 8F 48 8D 46 CF 44 4D 00 44 FF 00 48 EF :CC
5AA0 44 CF 46 4D 00 48 FF 00 43 4F 44 4F 46 4D 48 4D :3A
5AB0 46 8F 44 8D 00 43 FF 00 44 4F 46 4F 48 4D 49 4D :3B
5AC0 48 8F 48 8D 00 48 FF 48 FF 00 46 8F 44 FD 43 8F :22
5AD0 00 46 F1 00 38 8F 3F 8D 33 8D 3F 8D 00 38 8F 3F :5C
5AE0 8D 3C 8F 3F 8D 00 3C 8F 3F 8D 3A 8F 3F 8D 00 33 :83
5AF0 8F 3F 8D 3A 8F 3F 8D 00 38 8F 3F 8D 3C 8D 3F 8D :18
------------------------------------------------------------
Sum: BE 1B 50 81 37 7E 93 EE 3A B3 B3 87 BF 7F AE 8E :81
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5B00 00 38 8F 3F 8D 33 8F 3F 8D 00 32 8F 33 8D 3F 8F :70
5B10 3D 8D 00 3C 8F 33 8D 3C 8F 3A 8D 3A 8D 00 00 00 :AE
5B20 00 00 00 00 34 10 8E FC 80 17 00 06 A7 34 10 34 :8A
5B30 10 8E FC 80 17 00 06 A7 03 86 29 A7 02 8D 09 34 :03
5B40 02 86 80 B7 FD 05 35 82 7D FD 05 2B FB 39 34 02 :8C
5B50 B6 FC 80 8A 80 B7 FC 80 8D 02 35 82 7F FD 05 39 :6F
5B60 05 39 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :3E
5B70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5B80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5B90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5BA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5BB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5BC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5BD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5BE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5BF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
------------------------------------------------------------
Sum: 0A 0E 8B 3C E4 32 E1 20 A9 D6 22 23 E3 84 91 32 :E4
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5C00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5C10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5C20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5C30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5C40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5C50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5C60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5C70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5C80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5C90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5CA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5CB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5CC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5CD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5CE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5CF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
------------------------------------------------------------
Sum: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5D00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5D10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5D20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5D30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5D40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5D50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5D60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5D70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5D80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5D90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5DA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5DB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5DC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5DD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5DE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5DF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
------------------------------------------------------------
Sum: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5E00 20 05 20 0F 20 1A 00 8D 25 C0 04 4F 5D 2A 01 43 :1E
5E10 ED 02 39 8D 19 1F 89 4F 5A 5D 2A 01 43 ED 02 39 :12
5E20 8D 0C E6 A4 4F 5D 27 03 2A 01 43 ED 02 39 E6 03 :78
5E30 2A 01 50 A6 8C D0 4A 48 31 8D 00 16 10 AE A6 31 :78
5E40 A5 E6 A4 4F CB 0D 4C C0 09 27 02 2A F9 80 01 CB :03
5E50 09 39 5E 6E 5E 78 5E 80 5E 87 5E 8B 5E 8F 5E 92 :6D
5E60 5E 9C 5E A6 5E AE 5E B6 5E BE 5E C6 5E D4 08 F6 :8E
5E70 F7 F8 FF 01 08 09 0A 00 06 F7 FF 01 08 09 0A 00 :22
5E80 05 F6 F8 08 09 0A 00 02 11 13 00 01 09 01 09 01 :49
5E90 09 00 04 F7 FF 01 09 04 F7 FF 01 09 04 F6 F8 08 :0B
5EA0 0A 04 F6 F8 08 0A 06 F7 FF 01 08 09 0A 00 06 F7 :23
5EB0 FF 01 08 09 0A 00 06 F7 FF 01 08 09 0A 00 06 F7 :30
5EC0 FF 01 08 09 0A 00 08 F6 F7 F8 FF 01 08 09 0A 04 :27
5ED0 F7 FF 01 09 08 F6 F7 F8 FF 01 08 09 0A 04 F6 F8 :FA
5EE0 08 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :12
5EF0 00 00 00 00 00 00 41 53 4D 42 49 4F 32 00 07 00 :F4
------------------------------------------------------------
Sum: DC CC F1 5C CF AD 61 52 EE 5D 8F 44 D4 EE 14 F6 :0E
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5F00 20 0D 02 60 00 68 00 B0 00 68 00 80 00 B8 00 34 :7B
5F10 77 1A 10 B7 FD 0F 30 8C E9 E6 80 10 AE 81 EE 02 :9E
5F20 10 AC 84 24 06 A6 A0 A7 C0 20 F5 5A 27 04 30 04 :E5
5F30 20 E9 B6 FD 0F 35 F7 00 00 00 00 00 00 00 00 00 :F7
5F40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5F50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5F60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5F70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5F80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5F90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5FA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5FB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5FC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
```

リスト 2　ショウギSB

```
5FD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5FE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5FF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: C7 BC 4C 38 12 52 C7 E3 A9 6E 75 EA D5 3D 1E 3A :F5


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6000 00 18 7F E0 01 00 01 00 01 00 3F F8 01 00 01 00 :B3
6010 01 00 FF FE 01 00 01 00 01 00 01 00 01 00 0E 00 :11
6020 70 20 00 20 00 20 FB FE 00 20 00 20 F8 20 01 FC :1E
6030 00 00 F8 00 01 FC 01 04 F9 04 89 04 89 04 F9 FC :06
6040 00 00 FF FE 01 00 01 00 01 00 21 00 21 00 21 F8 :5B
6050 21 00 21 00 21 00 21 00 21 00 21 00 FF FE 00 00 :C3
6060 10 00 3E 7E 22 12 44 12 88 22 7F 4C 49 00 49 28 :85
6070 7F 7E 49 48 49 08 7F 7E 41 08 41 08 81 08 83 08 :82
6080 00 00 00 00 01 FC 8E 60 70 88 01 04 02 20 02 10 :1C
6090 04 00 04 00 04 00 04 00 02 00 01 F0 00 00 00 00 :03
60A0 01 80 00 80 00 84 83 FE 7C 80 00 80 07 80 08 80 :91
60B0 08 80 09 80 06 80 00 80 01 00 01 00 02 00 0C 00 :27
60C0 70 20 00 20 00 20 FB FE 00 20 00 20 F8 20 01 FC :1E
60D0 00 00 F8 00 01 FC 01 04 F9 04 89 04 89 04 F9 FC :06
60E0 00 00 03 80 1C 80 01 00 02 00 02 30 04 10 0F 10 :87
60F0 38 F0 48 18 48 24 30 20 00 40 00 80 03 00 00 00 :07
---------------------------------------------------------
Sum: D6 C6 6D 7A 00 F6 25 92 D0 BA 59 B8 00 FE 15 B8 :96


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6100 01 80 00 80 00 80 41 30 3F C0 01 00 09 00 11 30 :3C
6110 0F C0 01 00 01 00 0F 00 11 C0 11 20 0E 10 00 00 :00
6120 00 00 00 40 10 20 08 20 08 20 08 3E 8F C0 78 40 :0D
6130 08 40 08 40 08 C0 08 40 08 00 08 00 07 F0 00 00 :A7
6140 00 00 03 00 01 00 02 00 02 00 04 00 04 00 08 00 :18
6150 0B 00 1C 84 10 84 20 84 60 88 40 70 00 00 00 00 :7B
6160 00 00 3F F8 20 08 3F F8 20 08 3F F8 00 00 FF FE :F2
6170 22 00 3E FC 22 04 3E 44 22 28 2E 10 F2 28 02 C6 :6E
6180 40 00 41 FE 7C 00 50 00 90 FC 10 84 10 84 FE 84 :81
6190 10 84 10 FC 10 00 28 84 24 88 44 48 80 48 07 FE :61
61A0 00 18 7F E0 01 00 01 00 01 00 3F F8 01 00 01 00 :B3
61B0 01 00 FF FE 01 00 01 00 01 00 01 00 01 00 0E 00 :11
61C0 44 00 55 FE 54 20 54 20 54 FC 54 84 54 84 54 FC :CF
61D0 54 84 54 FC 54 84 54 84 54 FC 54 48 44 84 85 02 :13
61E0 00 00 00 00 01 FC 8E 60 70 88 01 04 02 20 02 10 :1C
61F0 04 00 04 00 04 00 04 00 02 00 01 F0 00 00 00 00 :03
---------------------------------------------------------
Sum: 32 A0 21 4A A7 90 B3 D8 D4 5C 11 5A CF DC 81 C4 :8A


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6200 00 00 00 60 40 20 20 20 24 2C 23 F0 20 20 40 20 :03
6210 40 20 40 20 50 20 51 E0 62 38 22 24 21 C0 00 00 :22
6220 04 00 02 00 02 30 22 C0 1F 00 04 00 07 C0 0C B0 :C0
6230 14 88 23 08 42 08 46 08 4A 08 38 10 01 E0 00 00 :DA
6240 00 00 10 00 08 00 09 C0 0A 20 12 10 12 10 14 10 :73
6250 14 10 14 10 08 10 08 20 00 20 00 40 01 80 06 00 :6F
6260 00 00 7F FC 01 00 01 00 01 00 01 00 01 00 3F F8 :B7
6270 01 00 01 00 01 00 01 00 01 00 01 00 FF FE 00 00 :03
6280 00 18 7F E0 01 00 01 00 01 00 3F F8 01 00 01 00 :B3
6290 01 00 FF FE 01 00 01 00 01 00 01 00 01 00 0E 00 :11
62A0 00 00 00 00 01 FC 8E 60 70 88 01 04 02 20 02 10 :1C
62B0 04 00 04 00 04 00 04 00 02 00 01 F0 00 00 00 00 :03
62C0 00 00 00 60 40 20 20 20 24 2C 23 F0 20 20 40 20 :03
62D0 40 20 40 20 50 20 51 E0 62 38 22 24 21 C0 00 00 :22
62E0 04 00 02 00 02 30 22 C0 1F 00 04 00 07 C0 0C B0 :C0
62F0 14 88 23 08 42 08 46 08 4A 08 38 10 01 E0 00 00 :DA
---------------------------------------------------------
Sum: CA 78 F0 FA C1 FC 59 D0 5E A0 58 84 A9 AE 02 B8 :FD


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6300 00 00 10 00 08 00 09 C0 0A 20 12 10 12 10 14 10 :73
6310 14 10 14 10 08 10 08 20 00 20 00 40 01 80 06 00 :6F
6320 06 20 F8 20 10 20 FE 20 10 FE FE 22 92 22 92 22 :22
6330 FE 22 92 22 92 22 FE 22 10 22 FE 42 10 42 FE 8C :F6
6340 04 00 02 00 02 00 04 00 07 00 58 B0 68 88 08 84 :97
6350 08 84 10 82 10 82 20 80 25 00 43 00 42 00 00 00 :FA
6360 00 00 00 40 10 20 08 20 08 20 08 3E 8F C0 78 40 :0D
6370 08 40 08 40 08 C0 08 40 08 00 08 00 07 F0 00 00 :A7
6380 00 48 00 44 00 40 7F FE 40 40 40 44 40 44 7E 48 :97
6390 42 48 42 50 42 20 42 20 4C 50 40 90 83 0A 80 06 :5F
63A0 10 00 08 00 08 00 0C E0 19 10 6A 10 0C 10 18 10 :F3
63B0 18 10 28 20 28 20 48 22 48 24 18 18 08 00 00 00 :C6
63C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
63D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
63E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
63F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: 96 B6 3A 08 4E 34 56 22 53 44 BB 9E CC 8A 40 E0 :EE


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6400 00 00 07 C0 08 20 10 10 10 10 10 10 10 10 10 10 :8F
6410 10 10 10 10 10 10 10 10 10 10 08 20 07 C0 00 00 :8F
6420 00 00 01 00 03 00 05 00 01 00 01 00 01 00 01 00 :0D
6430 01 00 01 00 01 00 01 00 01 00 01 00 07 C0 00 00 :CD
6440 00 00 07 C0 08 20 10 10 10 10 00 10 00 20 00 40 :9F
6450 01 80 02 00 04 00 08 00 10 00 10 10 1F F0 00 00 :CE
6460 00 00 07 C0 08 20 10 10 00 10 00 10 00 20 01 C0 :10
6470 00 20 00 10 00 10 00 10 10 10 08 20 07 C0 00 00 :5F
6480 00 00 00 80 01 80 02 80 02 80 04 80 04 80 08 80 :95
6490 10 80 10 80 20 80 3F F0 00 80 00 80 03 E0 00 00 :D2
64A0 00 00 1F E0 10 00 10 00 10 00 10 00 17 C0 18 20 :4E
64B0 10 10 00 10 00 10 00 10 10 10 08 20 07 C0 00 00 :5F
64C0 00 00 07 C0 08 20 10 10 10 00 10 00 17 C0 18 20 :3E
64D0 10 10 10 10 10 10 10 10 10 10 08 20 07 C0 00 00 :8F
64E0 00 00 1F F0 10 10 10 10 10 20 00 20 00 40 00 40 :1F
64F0 00 80 00 80 00 80 00 80 00 80 00 80 00 80 00 00 :80
---------------------------------------------------------
Sum: 42 D0 BE 90 89 50 CF 80 A4 10 66 60 88 A0 4A 10 :54


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6500 00 00 07 C0 08 20 10 10 10 10 10 10 08 20 07 C0 :3E
6510 08 20 10 10 10 10 10 10 10 10 08 20 07 C0 00 00 :97
6520 00 00 07 C0 08 20 10 10 10 10 10 10 10 10 10 10 :8F
6530 08 30 07 D0 00 10 00 10 10 10 08 20 07 C0 00 00 :3E
6540 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6550 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6560 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6570 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6580 03 C0 00 40 04 20 08 20 10 10 20 08 40 04 BF FA :94
6590 02 08 02 08 02 08 04 08 04 08 08 08 30 10 C0 60 :A6
65A0 0C 20 F0 20 10 20 11 24 11 22 FD 22 12 22 12 20 :59
65B0 30 24 38 24 54 68 90 08 10 10 10 20 10 C0 13 00 :37
65C0 10 00 11 FE FC 42 04 42 04 42 08 42 10 42 34 42 :FB
65D0 58 42 94 42 10 82 10 82 10 84 11 04 12 04 14 18 :7F
65E0 01 00 01 00 01 00 7F FC 41 04 41 04 41 04 41 04 :92
65F0 41 04 41 04 7F FC 41 04 01 00 01 00 01 00 01 00 :4E
---------------------------------------------------------
Sum: FB A2 36 30 16 D0 B1 58 CB 54 C0 FC 1C F0 45 A8 :C6


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6600 02 00 02 00 02 00 02 00 02 00 02 00 03 F8 02 00 :09
6610 02 00 02 00 02 00 02 00 02 00 02 00 FF FE 00 00 :09
6620 10 00 23 F8 C0 88 48 88 30 90 20 90 40 9C 49 44 :1C
6630 FD 44 21 44 A9 24 A9 28 A9 30 AA 10 A2 28 24 C6 :8B
6640 00 00 7E 7E 42 42 42 42 7E 7E 42 42 42 42 7E 7E :04
6650 40 02 47 E2 44 22 44 22 44 22 47 E2 44 22 40 06 :72
6660 00 00 3F F8 20 08 20 08 20 08 3F F8 20 08 20 08 :36
6670 20 08 3F F8 20 08 20 08 20 08 20 08 3F F8 20 08 :5E
6680 70 20 00 20 00 20 FB FE 00 20 00 20 F8 20 01 FC :1E
6690 00 00 F8 00 01 FC 01 04 F9 04 89 04 89 04 F9 FC :06
66A0 10 06 17 F8 10 00 94 84 52 44 32 48 10 10 10 28 :B5
66B0 30 08 57 FE 90 08 12 08 11 08 10 88 10 08 10 38 :50
66C0 21 08 21 08 27 FE F9 08 21 08 61 F8 61 08 B1 08 :1C
66D0 A9 F8 A9 08 21 08 27 FE 20 00 21 08 22 04 24 02 :35
66E0 82 10 41 20 2F FC 00 80 07 F8 E4 08 24 08 27 F8 :D4
66F0 24 08 24 08 27 F8 24 08 24 08 27 F8 50 00 8F FE :CB
---------------------------------------------------------
Sum: 91 94 20 DA 72 3E A1 40 A7 E8 0E B8 61 6E 12 F6 :DC


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6700 20 00 23 F8 22 08 23 F8 FA 08 23 F8 20 00 27 FE :E2
6710 21 00 23 FC 24 94 31 14 E6 24 00 C4 03 04 00 18 :2A
6720 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6730 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6740 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6750 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6760 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6770 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6780 00 00 01 00 01 00 01 00 01 00 01 00 3F F8 01 00 :3D
6790 01 00 01 00 01 00 01 00 01 00 01 00 FF FE 00 00 :03
67A0 00 00 7C FE 44 92 44 92 44 92 7C 92 44 92 44 FE :22
67B0 44 82 44 80 7C 80 44 80 44 80 44 82 84 82 8C 7E :E4
67C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FE :FD
67D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
67E0 01 00 01 00 01 00 7F FC 01 00 01 00 01 00 01 00 :82
67F0 FF FE 02 80 02 80 04 40 04 20 08 10 30 08 C0 06 :7F
---------------------------------------------------------
Sum: 86 80 0B F2 0B 2E 61 5A 6F 5E EE E0 5A 16 B8 96 :50


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6800 00 00 00 00 00 00 00 00 03 C0 00 00 00 00 3F FC :FE
6810 00 00 00 03 FF FF C0 00 00 3F FF FF FC 00 00 FC :F6
6820 00 00 3F 00 00 FF FE 7F FF 00 01 FF FE 7F FF 80 :B6
6830 01 FF 00 00 FF 80 03 FF FE 7F 3F C0 03 FF FE 7F :7C
6840 9F C0 07 C0 00 00 03 E0 07 FF FF FF FF E0 0F F3 :EE
6850 9C 00 3F F0 0F F0 19 9E 7F F0 1F FF 9C C9 CF F8 :3A
6860 1F C0 10 00 03 F8 3F F3 9F E7 CF FC 3F E7 9F FE :30
6870 0F FC 7F FF FF FF FF FE FF FF FC 3F FF FF FF FF :B9
6880 C3 C3 FF FF FF FC 3F FC 3F FF FF C3 FF FF C3 FF :7A
6890 FF 3F FF FF FC FF FE FC 00 00 3F 7F FE FF FE 7F :69
68A0 FF 7F FD FF FE 7F FF BF FD FF 00 00 FF BF FB FF :69
68B0 FE 7F 3F DF FB FF FE 7F 9F DF F7 C0 00 00 03 EF :39
68C0 F7 FF FF FF FF EF EF F3 9C 00 3F F7 EF F0 19 9E :2C
68D0 7F F7 DF FF 9C C9 CF FB DF C0 10 00 03 FB BF F3 :E2
68E0 9F E7 CF FD BF E7 9F FE 0F FD 7F FF FF FF FF FE :1A
68F0 55 55 54 15 55 55 AA AA 83 C2 AA AA 55 54 3F FC :8E
---------------------------------------------------------
Sum: 93 AD 4F 9E B2 D2 5C B9 0C AF D5 99 18 08 8D D6 :72


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6900 15 55 AA 83 FF FF C2 AA 55 3F FF FF FC 55 AA FC :8A
6910 00 00 3F 2A 54 FF FE 7F FF 55 A9 FF FE 7F FF AA :5B
6920 55 FF 00 00 FF 95 AB FF FE 7F 3F CA 53 FF FE 7F :E7
6930 9F D5 A7 C0 00 00 03 EA 57 FF FF FF FF E5 AF F3 :A2
6940 9C 00 3F F2 4F F0 19 9E 7F F5 9F FF 9C C9 CF FA :03
```

リスト 2　ショウギSB

```
6950 5F C0 10 00 03 F9 BF F3 9F E7 CF FC 3F E7 9F FE :F1
6960 0F FD 7F FF FF FF FF FE 00 00 00 00 00 00 00 00 :85
6970 03 C0 00 00 00 00 3C 3C 00 00 00 03 E1 87 C0 00 :66
6980 00 3F 0F F0 FC 00 00 F8 FF FF 1F 00 00 E3 00 00 :32
6990 C7 00 01 FF FE 7F FF 80 01 FF 80 01 FF 80 03 FF :C5
69A0 9E 79 FF C0 03 FF E6 67 FF C0 07 F8 00 00 3F E0 :02
69B0 07 FF FF FF FF E0 0F F3 9C 00 3F F0 0F F0 19 9E :66
69C0 7F F0 1F FF 9C C9 CF F8 1F C0 10 00 03 F8 3F F3 :D5
69D0 9F CF CF FC 3F E7 9F FE 0F FC 7F FF FF FF FF FE :80
69E0 FF FF FC 3F FF FF FF FF C3 C3 FF FF FF FC 3C 3C :2C
69F0 3F FF FF C3 E1 87 C3 FF FF 3F 0F F0 FC FF FE F8 :58
-----------------------------------------------------------------
Sum: DE 1A 55 09 5A 0F A5 A3 52 6A D6 9C 13 34 57 B2 :85


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6A00 FF FF 1F 7F FE E3 00 00 C7 7F FD FF FE 7F FF BF :FA
6A10 FD FF 80 01 FF BF FB FF 9E 79 FF DF FB FF E6 67 :71
6A20 FF DF F7 F8 00 00 3F EF F7 FF FF FF FF EF EF F3 :BF
6A30 9C 00 3F F7 EF F0 19 9E 7F F7 DF FF 9C C9 CF FB :EB
6A40 DF C0 10 00 03 FB BF F3 9F CF CF FD BF E7 9F FE :DC
6A50 0F FD 7F FF FF FF FF FE 55 55 54 15 55 55 AA AA :96
6A60 83 C2 AA AA 55 54 3C 3C 15 55 AA 83 E1 87 C2 AA :25
6A70 55 3F 0F F0 FC 55 AA F8 FF FF 1F 2A 54 E3 00 00 :04
6A80 C7 55 A9 FF FE 7F FF AA 55 FF 80 01 FF 95 AB FF :FD
6A90 9E 79 FF CA 53 FF E6 67 FF D5 A7 F8 00 00 3F EA :1B
6AA0 57 FF FF FF FF E5 AF F3 9C 00 3F F2 4F F0 19 9E :9D
6AB0 7F F5 9F FF 9C C9 CF FA 5F C0 10 00 03 F9 BF F3 :1D
6AC0 9F CF CF FC 3F E7 9F FE 0F FD 7F FF FF FF FF FE :81
6AD0 00 00 00 00 00 00 00 00 03 C0 00 00 00 00 3F FC :FE
6AE0 00 00 00 03 E3 FF C0 00 00 3F 19 80 3C 00 00 FC :B5
6AF0 FC 9F 3F 00 00 F2 01 80 3F 00 01 FF CF 9F 3F 80 :B9
-----------------------------------------------------------------
Sum: 33 CB 71 CE 4D 39 BA 2D 83 F6 D5 04 38 F8 ED 56 :6F


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6B00 01 FE 01 80 3F 80 03 FF CF 93 CF C0 03 FE 49 9C :18
6B10 3F C0 07 F8 01 81 81 E0 07 FF FF FF FF E0 0F E6 :B9
6B20 70 00 1F F0 0F E0 73 3E 7F F0 1F FE 79 99 CF F8 :84
6B30 1F 80 60 00 01 F8 3F E6 7F CF CF FC 3F E6 7F FE :D8
6B40 0F FC 7F FF FF FF FF FE FF FF FC 3F FF FF FF FF :B9
6B50 C3 C3 FF FF FF FC 3F FC 3F FF FF C3 E3 FF C3 FF :5E
6B60 FF 3F 19 80 3C FF FE FC FC 9F 3F 7F FE F2 01 80 :D6
6B70 3F 7F FD FF CF 9F 3F BF FD FE 01 80 3F BF FB FF :9A
6B80 CF 93 CF DF FB FE 49 9C 3F DF F7 F8 01 81 81 EF :ED
6B90 F7 FF FF FF FF EF EF E6 70 00 1F F7 EF E0 73 3E :BD
6BA0 7F F7 DF FE 79 99 CF FB DF 80 60 00 01 FB BF E6 :8F
6BB0 7F CF CF FD BF E6 7F FE 0F FD 7F FF FF FF FF FE :C1
6BC0 55 55 54 15 55 55 AA AA 83 C2 AA AA 55 54 3F FC :8E
6BD0 15 55 AA 83 E3 FF C2 AA 55 3F 19 80 3C 55 AA FC :49
6BE0 FC 9F 3F 2A 54 F2 01 80 3F 55 A9 FF CF 9F 3F AA :5E
6BF0 55 FE 01 80 3F 95 AB FF CF 93 CF CA 53 FE 49 9C :83
-----------------------------------------------------------------
Sum: 5E 5A D5 00 56 B9 4F 06 8E 31 27 9B 7C AD 87 44 :66


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6C00 3F D5 A7 F8 01 81 81 EA 57 FF FF FF FF E5 AF E6 :6D
6C10 70 00 1F F2 4F E0 73 3E 7F F5 9F FE 79 99 CF FA :4D
6C20 5F 80 60 00 01 F9 BF E6 7F CF CF FC 3F E6 7F FE :99
6C30 0F FD 7F FF FF FF FF FE 00 00 00 00 00 00 00 00 :85
6C40 03 C0 00 00 00 00 3F FC 00 00 00 03 FF FF C0 00 :BF
6C50 00 3F 9F F3 FC 00 00 F8 03 80 7F 00 00 FF 0F F3 :C8
6C60 FF 00 01 FE 02 00 1F 80 01 FC 93 F3 FF 80 03 F9 :9D
6C70 9F 00 3F C0 03 E3 9F F3 FF C0 07 FF 9C 00 07 E0 :5E
6C80 07 FF FF FF FF E0 0F FE 00 03 FF F0 0F FE 7C 7F :EA
6C90 FF F0 1F FE 7C 7F FF F8 1F FE 00 00 3F F8 3F FF :90
6CA0 FF FF 9F FC 3F F8 00 01 9F FC 7F FF FF FF FF FE :E5
6CB0 FF FF FC 3F FF FF FF FF C3 C3 FF FF FF FC 3F FC :EF
6CC0 3F FF FF C3 FF FF C3 FF FF 3F 9F F3 FC FF FE F8 :81
6CD0 03 80 7F 7F FE FF 0F F3 FF 7F FD FE 02 00 1F BF :D9
6CE0 FD FC 93 F3 FF BF FB F9 9F 00 3F DF FB E3 9F F3 :5E
6CF0 FF DF F7 FF 9C 00 07 EF F7 FF FF FF FF EF EF FE :35
-----------------------------------------------------------------
Sum: 00 98 45 06 A2 4F 90 43 6D 7C DD AB 95 A4 7A CA :95


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6D00 00 03 FF F7 EF FE 7C 7F FF F7 DF FE 7C 7F FF FB :A9
6D10 DF FE 00 00 3F FB BF FF FF FF 9F FD BF F8 00 01 :27
6D20 9F FD 7F FF FF FF FF FE 55 55 54 15 55 55 AA AA :26
6D30 83 C2 AA AA 55 54 3F FC 15 55 AA 83 FF FF C2 AA :7E
6D40 55 3F 9F F3 FC 55 AA F8 03 80 7F 2A 54 FF 0F F3 :9A
6D50 FF 55 A9 FE 02 00 1F AA 55 FC 93 F3 FF 95 AB F9 :D5
6D60 9F 00 3F CA 53 E3 9F F3 FF D5 A7 FF 9C 00 07 EA :77
6D70 57 FF FF FF FF E5 AF FE 00 03 FF F2 4F FE 7C 7F :21
6D80 FF F5 9F FE 7C 7F FF FA 5F FE 00 00 3F F9 BF FF :D8
6D90 FF FF 9F FC 3F F8 00 01 9F FD 7F FF FF FF FF FE :E6
6DA0 00 00 00 00 00 00 00 00 03 C0 00 00 00 00 3F FC :FE
6DB0 00 00 00 03 F8 0F C0 00 00 3F FE 7F FC 00 00 FE :80
6DC0 00 00 7F 00 00 FF C6 63 FF 00 01 FF 00 00 FF 80 :25
6DD0 01 FC 1F F8 3F 80 03 E1 80 01 83 C0 03 FF 9F F9 :15
6DE0 FF C0 07 FF 80 01 FF E0 07 FF FE 7F FF E0 0F FE :94
6DF0 00 00 7F F0 0F FF FE 7F FF F0 1F FE 00 00 7F F8 :7D
-----------------------------------------------------------------
Sum: 49 03 10 3E 53 6E 15 A9 45 DE 52 5B 09 34 D1 0B :02


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6E00 1F FE 00 00 7F F8 3F FF FE 7F FF FC 3F F8 00 00 :81
6E10 0F FC 7F FF FE 7F FF FE FF FF FC 3F FF FF FF FF :38
6E20 C3 C3 FF FF FF FC 3F FC 3F FF FF C3 F8 0F C3 FF :83
6E30 FF 3F FE 7F FC FF FE FE 00 00 7F 7F FE FF C6 63 :D6
6E40 FF 7F FD FF 00 00 FF BF FD FC 1F F8 3F BF FB E1 :22
6E50 80 01 83 DF FB FF 9F F9 FF DF F7 FF 80 01 FF EF :B8
6E60 F7 FF FE 7F FF EF EF FE 00 00 7F F7 EF FF FE 7F :2F
6E70 FF F7 DF FE 00 00 7F FB DF FE 00 00 7F FB BF FF :62
6E80 FE 7F FF FD BF F8 00 00 0F FD 7F FF FE 7F FF FE :34
6E90 55 55 54 15 55 55 AA AA 83 C2 AA AA 55 54 3F FC :8E
6EA0 15 55 AA 83 F8 0F C2 AA 55 3F FE 7F FC 55 AA FE :14
6EB0 00 00 7F 2A 54 FF C6 63 FF 55 A9 FF 00 00 FF AA :CA
6EC0 55 FC 1F F8 3F 95 AB E1 80 01 83 CA 53 FF 9F F9 :80
6ED0 FF D5 A7 FF 80 01 FF EA 57 FF FE 7F FF E5 AF FE :48
6EE0 00 00 7F F2 4F FF FE 7F FF F5 9F FE 00 00 7F FA :46
6EF0 5F FE 00 00 7F F9 BF FF FE 7F FF FC 3F F8 00 00 :42
-----------------------------------------------------------------
Sum: 80 6A 9A 80 5F 49 20 A8 D1 1D FD D5 41 C3 F3 42 :6D


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6F00 0F FD 7F FF FE 7F FF FE 00 00 00 00 00 00 00 00 :04
6F10 03 C0 00 00 00 00 3F FC 00 00 00 03 FE 7F C0 00 :3E
6F20 00 3F CE 01 FC 00 00 FF CE 7F FF 00 00 F8 00 00 :4D
6F30 1F 00 01 FF 8E 79 FF 80 01 FE 3E 7C 7F 80 03 F0 :50
6F40 F8 63 1F C0 03 FF FF 0F FF C0 07 FF 80 7F FF E0 :ED
6F50 07 FF E7 FF FF E0 0F FF 8F FF FF F0 0F FE 00 00 :63
6F60 7F F0 1F FE 7F 9F FF F8 1F FE 7F CF FF F8 3F 80 :C2
6F70 00 00 07 FC 3F FF 3F 9F FF FC 7F F8 7F C3 FF FE :D0
6F80 FF FF FC 3F FF FF FF FF C3 C3 FF FF FF FC 3F FC :EF
6F90 3F FF FF C3 FE 7F C3 FF FF 3F CE 01 FC FF FE FF :44
6FA0 CE 7F FF 7F FE F8 00 00 1F 7F FD FF 8E 79 FF BF :20
6FB0 FD FE 3E 7C 7F BF FB F0 F8 63 1F DF FB FF FF 0F :3F
6FC0 FF DF F7 FF 80 7F FF EF F7 FF E7 FF FF EF EF FF :79
6FD0 8F FF FF F7 EF FE 00 00 7F F7 DF FE 7F 9F FF FB :DC
6FE0 DF FE 7F CF FF FB BF 80 00 00 07 FD BF FF 3F 9F :04
6FF0 FF FD 7F F8 7F C3 FF FE 55 55 54 15 55 55 AA AA :C3
-----------------------------------------------------------------
Sum: 24 A2 A6 72 AF E5 03 79 1F 65 4B 22 A0 84 12 5A :6F


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7000 83 C2 AA AA 55 54 3F FC 15 55 AA 83 FE 7F C2 AA :FD
7010 55 3F CE 01 FC 55 AA FF CE 7F FF 2A 54 F8 00 00 :1F
7020 1F 55 A9 FF 8E 79 FF AA 55 FE 3E 7C 7F 95 AB F0 :88
7030 F8 63 1F CA 53 FF FF 0F FF D5 A7 FF 80 7F FF EA :06
7040 57 FF E7 FF FF E5 AF FF 8F FF FF F2 4F FE 00 00 :9A
7050 7F F5 9F FE 7F 9F FF FA 5F FE 7F CF FF F9 BF 80 :0A
7060 00 00 07 FC 3F FF 3F 9F FF FD 7F F8 7F C3 FF FE :D1
7070 00 00 00 00 00 00 00 00 03 C0 00 00 00 00 3F FC :FE
7080 00 00 00 03 FF FF C0 00 00 3F 80 01 3C 00 00 F0 :AD
7090 3C 7C CF 00 00 FE 7C 7E 39 00 01 F0 00 01 87 80 :B1
70A0 01 FE 7C 79 1F 80 03 F8 FC 78 C4 C0 03 E3 FC 7C :E4
70B0 01 C0 07 FF FE FF FF E0 07 F8 00 00 1F E0 0F FF :AF
70C0 FC 7F FF F0 0F F8 00 00 1F F0 1F F9 FC 7F 9F F8 :AA
70D0 1F F8 00 00 1F F8 3F FF FC 7F FF FC 3F E0 00 00 :01
70E0 07 FC 7F FF FC 7F FF FE FF FF FC 3F FF FF FF FF :2E
70F0 C3 C3 FF FF FF FC 3F FC 3F FF FF C3 FF FF C3 FF :7A
-----------------------------------------------------------------
Sum: E8 1D 9C D6 34 8B 8F 9B BC 7D E9 89 B5 66 5C DF :61


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7100 FF 3F 80 01 3C FF FE F0 3C 7C CF 7F FE FE 7C 7E :E4
7110 39 7F FD F0 00 01 87 BF FD FE 7C 79 1F BF FB F8 :AD
7120 FC 78 C4 DF FB E3 FC 7C 01 DF F7 FF FE FF FF EF :2E
7130 F7 F8 00 00 1F EF EF FF FC 7F FF F7 EF F8 00 00 :43
7140 1F F7 DF F9 FC 7F 9F FB DF F8 00 00 1F FB BF FF :B2
7150 FC 7F FF FD BF E0 00 00 07 FD 7F FF FC 7F FF FE :10
7160 55 55 54 15 55 55 AA AA 83 C2 AA AA 55 54 3F FC :8E
7170 15 55 AA 83 FF FF C2 AA 55 3F 80 01 3C 55 AA F0 :41
7180 3C 7C CF 2A 54 FE 7C 7E 39 55 A9 F0 00 01 87 AA :56
7190 55 FE 7C 79 1F 95 AB F8 FC 78 C4 CA 53 E3 FC 7C :4F
71A0 01 D5 A7 FF FE FF FF EA 57 F8 00 00 1F E5 AF FF :63
71B0 FC 7F FF F2 4F F8 00 00 1F F5 9F F9 FC 7F 9F FA :73
71C0 5F F8 00 00 1F F9 BF FF FC 7F FF FC 3F E0 00 00 :C2
71D0 07 FD 7F FF FC 7F FF FE 00 00 00 00 00 00 00 00 :FA
71E0 03 C0 00 00 00 00 31 FC 00 00 00 03 E0 07 C0 00 :9A
71F0 00 3F 87 8F FC 00 00 FC 60 00 7F 00 00 FF C7 1C :0E
-----------------------------------------------------------------
Sum: A7 10 14 80 3C 87 90 CE FB 07 74 4A 43 05 75 89 :72


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7200 7F 00 01 FF C0 00 7F 80 01 FF C7 1C 7F 80 03 FF :22
7210 C0 00 7F C0 03 FF 8F FC 7F C0 07 F8 1F F0 7F E0 :38
7220 07 FF FF FF FF E0 0F E1 86 00 07 F0 0F 9E 1F FF :1B
7230 FF F0 1C 70 FC 00 01 F8 1F 88 FF F8 FF F8 3F F8 :3C
7240 FF F8 FF FC 3F F8 FF F0 FF FC 7F FF FF FF FF FE :8C
7250 FF FF FC 3F FF FF FF FF C3 C3 FF FF FF FC 31 FC :E1
7260 3F FF FF C3 E0 07 C3 FF FF 3F 87 8F FC FF FE FC :F2
7270 60 00 7F 7F FE FF C7 1C 7F 7F FD FF C0 00 7F BF :36
7280 FD FF C7 1C 7F BF FB FF C0 00 7F DF FB FF 8F FC :BA
7290 7F DF F7 F8 1F F0 7F EF F7 FF FF FF FF EF EF E1 :7C
72A0 86 00 07 F7 EF 9E 1F FF FF F7 DC 70 FC 00 01 FB :69
72B0 DF 88 FF F8 FF FB BF F8 FF F8 FF FD BF F8 FF F0 :A8
72C0 FF FD 7F FF FF FF FF FE 55 55 54 15 55 55 AA AA :86
72D0 83 C2 AA AA 55 54 31 FC 15 55 AA 83 E0 07 C2 AA :59
72E0 55 3F 87 8F FC 55 AA FC 60 00 7F 2A 54 FF C7 1C :E0
72F0 7F 55 A9 FF C0 00 7F AA 55 FF C7 1C 7F 95 AB FF :5A
-----------------------------------------------------------------
Sum: 19 9E 31 E5 76 CC 57 E4 39 5B 73 B1 23 D6 E9 C2 :A6
```

リスト 2　ショウギSB

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7300 C0 00 7F CA 53 FF 8F FC 7F D5 A7 F8 1F F0 7F EA :51
7310 57 FF FF FF FF E5 AF E1 86 00 07 F2 4F 9E 1F FF :52
7320 FF F5 9C 70 FC 00 01 FA 5F 88 FF F8 FF F9 BF F8 :84
7330 FF F8 FF FC 3F F8 FF F0 FF FD 7F FF FF FF FF FE :8D
7340 00 00 00 00 00 00 00 00 03 C0 00 00 00 00 3C 7C :7B
7350 00 00 00 03 FE 33 C0 00 00 3C 00 00 FC 00 00 FC :28
7360 3F 8F 0F 00 00 F8 87 C4 3F 00 01 E1 E3 E0 F1 80 :75
7370 01 87 03 0C 67 80 03 FF FF FF 0F C0 03 FF 0F FF :5D
7380 FF C0 07 FC 63 80 1F E0 07 E1 F8 9F 9F E0 0F 0C :BD
7390 03 80 1F F0 0F FF 9F 9F 9F F0 1F FC 03 80 1C F8 :1F
73A0 1F FF 9F 8C E1 F8 3F FC 93 8F 0F FC 3F E0 01 80 :2A
73B0 E0 3C 7F FF FF FF FF FE FF FF FC 3F FF FF FF FF :CA
73C0 C3 C3 FF FF FF FC 3F FC 3F FF FF C3 FF FF C3 FF :7A
73D0 FF 3F FF FF FC FF FE FF FF FF FF 7F FE FF FF FF :AB
73E0 FF 7F FD FF FF FF FF BF FD FF FF FF FF BF FB FF :E8
73F0 FF FF FF DF FB FF FF FF FF DF F7 FF FF FF FF EF :94
-----------------------------------------------------------
Sum: 16 FD 69 97 39 F6 BF BC 16 90 52 98 29 60 7F 45 :9A


Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7400 F7 FF FF FF FF EF EF FF FF FF FF F7 EF FF FF FF :B0
7410 FF F7 DF FF FF FF FF FB DF FF FF FF FF FB BF FF :60
7420 FF FF FF FD BF FF FF FF FF FD 7F FF FF FF FF FE :2B
7430 55 55 54 15 55 55 AA AA 83 C2 AA AA 55 54 3C 7C :0B
7440 15 55 AA 83 FE 33 C2 AA 55 3C 00 00 FC 55 AA FC :BC
7450 3F 8F 0F 2A 54 F8 87 C4 3F 55 A9 E1 E3 E0 F1 AA :1A
7460 55 87 03 0C 67 95 AB FF FF FF 0F CA 53 FF 0F FF :C8
7470 FF D5 A7 FC 63 80 1F EA 57 E1 F8 9F 9F E5 AF 0C :71
7480 03 80 1F F2 4F FF 9F 9F 9F F5 9F FC 03 80 1C FA :E8
7490 5F FF 9F 8C E1 F9 BF FC 93 8F 0F FC 3F E0 01 80 :EB
74A0 E0 3D 7F FF FF FF FF FE 00 00 00 00 00 00 00 00 :96
74B0 03 C0 00 00 00 00 3C 7C 00 00 00 03 FE 33 C0 00 :6F
74C0 00 3C 00 00 FC 00 00 FC 3F 8F 0F 00 00 F8 87 C4 :54
74D0 3F 00 01 E1 E3 E0 F1 80 01 87 03 0C 67 80 03 FF :D5
74E0 FF FF 0F C0 03 FE 7F F9 FF C0 07 C0 07 F9 FF E0 :AB
74F0 07 FC 3F 00 0F E0 0F F8 1F F9 FF F0 0F F2 4E 00 :8E
-----------------------------------------------------------
Sum: 7C 3D 20 E3 4E 37 C2 7C DA 81 9D A0 D0 5C 06 46 :8F


Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7500 07 F0 1F E6 67 F9 FF F8 1F CE 7F 00 0F F8 3F 1E :23
7510 7F F9 FF FC 3F FE 7C 00 03 FC 7F FF FF FF FF FE :A4
7520 FF FF FC 3F FF FF FF FF C3 C3 FF FF FF FC 3F FC :EF
7530 3F FF FF C3 FF FF C3 FF FF 3F FF FF FC FF FE FF :F4
7540 FF FF FF 7F FE FF FF FF FF 7F FD FF FF FF FF BF :AD
7550 FD FF FF FF FF BF FB FF FF FF FF DF FB FF FF FF :86
7560 FF DF F7 FF FF FF FF EF F7 FF FF FF FF EF EF FF :90
7570 FF FF FF F7 EF FF FF FF FF F7 DF FF FF FF FF FB :AC
7580 DF FF FF FF FF FB BF FF FF FF FF FD BF FF FF FF :4A
7590 FF FD 7F FF FF FF FF FE 55 55 54 15 55 55 AA AA :86
75A0 83 C2 AA AA 55 54 3C 7C 15 55 AA 83 FE 33 C2 AA :2E
75B0 55 3C 00 00 FC 55 AA FC 3F 8F 0F 2A 54 F8 87 C4 :26
75C0 3F 55 A9 E1 E3 E0 F1 AA 55 87 03 0C 67 95 AB FF :0D
75D0 FF FF 0F CA 53 FE 7F F9 FF D5 A7 C0 07 F9 FF EA :C4
75E0 57 FC 3F 00 0F E5 AF F8 1F F9 FF F2 4F F2 4E 00 :C5
75F0 07 F5 9F E6 67 F9 FF FA 5F CE 7F 00 0F F9 BF 1E :6B
-----------------------------------------------------------
Sum: 10 02 CB 91 8A 10 F7 EC 52 9B 0A 56 33 D6 10 ED :3E


Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7600 7F F9 FF FC 3F FE 7C 00 03 FD 7F FF FF FF FF FE :A5
7610 00 00 00 00 00 00 00 00 03 C0 00 00 00 00 3C 7C :7B
7620 00 00 00 03 FE 33 C0 00 00 3C 00 00 FC 00 00 FC :28
7630 3F 8F 0F 00 00 F8 87 C4 3F 00 01 E1 E3 E0 F1 80 :75
7640 01 87 03 0C 67 80 03 FF FF FF 0F C0 03 FF 80 07 :D6
7650 FF C0 07 FF FC FF FF E0 07 F0 00 00 1F E0 0F FF :A3
7660 0C C3 FF F0 0F F8 00 00 FF F0 1F C1 FF FC 1F F8 :A6
7670 18 18 00 00 C1 F8 3F F9 FF FC FF FC 3F F8 00 00 :4E
7680 FF FC 7F FF FF FF FF FE FF FF FC 3F FF FF FF FF :A9
7690 C3 C3 FF FF FF FC 3F FC 3F FF FF C3 FF FF C3 FF :7A
76A0 FF 3F FF FF FC FF FE FF FF FF FF 7F FE FF FF FF :AB
76B0 FF 7F FD FF FF FF FF BF FD FF FF FF FF BF FB FF :E8
76C0 FF FF FF DF FB FF FF FF FF DF F7 FF FF FF FF EF :94
76D0 F7 FF FF FF FF EF EF FF FF FF FF F7 EF FF FF FF :B0
76E0 FF F7 DF FF FF FF FF FB DF FF FF FF FF FB BF FF :60
76F0 FF FF FF FD BF FF FF FF FF FD 7F FF FF FF FF FE :2B
-----------------------------------------------------------
Sum: 96 1B 6D D0 21 7D 2B 4C 5F AA 1A D1 25 66 52 DB :AF


Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7700 55 55 54 15 55 55 AA AA 83 C2 AA AA 55 54 3C 7C :0B
7710 15 55 AA 83 FE 33 C2 AA 55 3C 00 00 FC 55 AA FC :BC
7720 3F 8F 0F 2A 54 F8 87 C4 3F 55 A9 E1 E3 E0 F1 AA :1A
7730 55 87 03 0C 67 95 AB FF FF FF 0F CA 53 FF 80 07 :41
7740 FF D5 A7 FF FC FF FF EA 57 F0 00 00 1F E5 AF FF :57
7750 0C C3 FF F2 4F F8 00 00 FF F5 9F C1 FF FC 1F FA :6F
7760 58 18 00 00 C1 F9 BF F9 FF FC FF FC 3F F8 00 00 :0F
7770 FF FD 7F FF FF FF FF FE 00 00 00 00 00 00 00 00 :75
7780 03 C0 00 00 00 00 3F FC 00 00 00 03 FF FF C0 00 :BF
7790 00 3F E7 FF FC 00 00 FF C3 FF FF 00 00 FF C3 FF :A2
77A0 FF 00 01 FF C3 FF FF 80 01 FF E1 FE 1F 80 03 FF :C0
77B0 F0 F8 1F C0 03 FF F8 41 FF C0 07 FF FC 1F FF E0 :C1
77C0 07 FF E0 FF FF E0 0F FF 07 FF FF F0 0F FE 1F FF :F2
77D0 FF F0 1F FE 0F FE 0F F8 1F FF 00 00 0F F8 3F FF :83
77E0 C0 00 7F FC 3F FF FF FF FF FC 7F FF FF FF FF FE :EB
77F0 FF FF FC 3F FF FF FF FF C3 C3 FF FF FF FC 3F FC :EF

-----------------------------------------------------------
Sum: 17 52 B6 B4 27 DE AD A9 16 AE 64 00 1A EF 46 F8 :9D


Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7800 3F FF FF C3 FF FF C3 FF FF 3F FF FF FC FF FE FF :F4
7810 FF FF FF 7F FE FF FF FF FF 7F FD FF FF FF FF BF :AD
7820 FD FF FF FF FF BF FB FF FF FF FF DF FB FF FF FF :86
7830 FF DF F7 FF FF FF FF EF F7 FF FF FF FF EF EF FF :90
7840 FF FF FF F7 EF FF FF FF FF F7 DF FF FF FF FF FB :AC
7850 DF FF FF FF FF FB BF FF FF FF FF FD BF FF FF FF :4A
7860 FF FD 7F FF FF FF FF FE 55 55 54 15 55 55 AA AA :86
7870 83 C2 AA AA 55 54 3F FC 15 55 AA 83 FF FF C2 AA :7E
7880 55 3F E7 FF FC 55 AA FF C3 FF FF 2A 54 FF C3 FF :74
7890 FF 55 A9 FF C3 FF FF AA 55 FF E1 FE 1F 95 AB FF :F8
78A0 F0 F8 1F CA 53 FF F8 41 FF D5 A7 FF FC 1F FF EA :DA
78B0 57 FF E0 FF FF E5 AF FF 07 FF FF F2 4F FE 1F FF :29
78C0 FF F5 9F FE 0F FE 0F FA 5F FF 00 00 0F F9 BF FF :CB
78D0 C0 00 7F FC 3F FF FF FF FF FD 7F FF FF FF FF FE :EC
78E0 00 00 00 00 00 00 00 00 03 C0 00 00 00 00 3F FC :FE
78F0 00 00 00 03 FF FF C0 00 00 3F FF FF FC 00 00 FE :F8
-----------------------------------------------------------
Sum: F4 19 C8 A3 9B 3D D6 C6 DB 29 DA 87 CF E7 DE E8 :CD


Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7900 3F CF FF 00 00 C0 01 C0 0F 00 01 F9 E7 CF FF 80 :CC
7910 01 00 00 C0 1F 80 03 FF FF FF 9F C0 03 E0 03 C0 :65
7920 1F C0 07 E7 F3 CF FF E0 07 E0 03 C8 0F E0 0F E7 :05
7930 F3 CF FF F0 0F E0 03 C8 0F F0 1F E7 F3 CF FF F8 :29
7940 1F E7 F3 C8 0F F8 3F E7 F3 CF F1 FC 3F 8F C3 C0 :EE
7950 03 FC 7F FF FF FF FF FE FF FF FC 3F FF FF FF FF :AD
7960 C3 C3 FF FF FF FC 3F FC 3F FF FF C3 FF FF C3 FF :7A
7970 FF 3F FF FF FC FF FE FF FF FF FF 7F FE FF FF FF :AB
7980 FF 7F FD FF FF FF FF BF FD FF FF FF FF BF FB FF :E8
7990 FF FF FF DF FB FF FF FF FF DF F7 FF FF FF FF EF :94
79A0 F7 FF FF FF FF EF EF FF FF FF FF F7 EF FF FF FF :B0
79B0 FF F7 DF FF FF FF FF FB DF FF FF FF FF FB BF FF :60
79C0 FF FF FF FD BF FF FF FF FF FD 7F FF FF FF FF FE :2B
79D0 55 55 54 15 55 55 AA AA 83 C2 AA AA 55 54 3F FC :8E
79E0 15 55 AA 83 FF FF C2 AA 55 3F FF FF FC 55 AA FE :8C
79F0 3F CF FF 2A 54 C0 01 C0 0F 55 A9 F9 E7 CF FF AA :71
-----------------------------------------------------------
Sum: D2 2F 4B F7 89 E0 D9 12 14 CA 72 7A 4A 19 33 6A :61


Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7A00 55 00 00 C0 1F 95 AB FF FF FF 9F CA 53 E0 03 C0 :D0
7A10 1F D5 A7 E7 F3 CF FF EA 57 E0 03 C8 0F E5 AF E7 :B9
7A20 F3 CF FF F2 4F E0 03 C8 0F F5 9F E7 F3 CF FF FA :F2
7A30 5F E7 F3 C8 0F F9 BF E7 F3 CF F1 FC 3F 8F C3 C0 :AF
7A40 03 FD 7F FF FF FF FF FE 00 00 00 00 00 00 00 00 :79
7A50 03 C0 00 00 00 00 3F FC 00 00 00 03 FF FF C0 00 :BF
7A60 00 3E 00 00 7C 00 00 FE 3F 1F FF 00 00 FE 3F 1F :71
7A70 FF 00 01 FE 00 00 FF 80 01 FE 3F 1F FF 80 03 FE :5A
7A80 3F 1F FF C0 03 FE 00 00 FF C0 07 FE 3F 1F FF E0 :1F
7A90 07 FE 3F 1F FF E0 0F FE 00 00 03 F0 0F FF FF FF :4E
7AA0 F8 F0 1F FC 63 18 FC 78 1F F0 F1 8C 7C 78 3F E3 :94
7AB0 F1 CE 38 FC 3E 0F F8 C7 11 FC 7F FF FF FF FF FE :85
7AC0 FF FF FC 3F FF FF FF FF C3 C3 FF FF FF FC 3F FC :EF
7AD0 3F FF FF C3 FF FF C3 FF FF 3F FF FF FC FF FE FF :F4
7AE0 FF FF FF 7F FE FF FF FF FF 7F FD FF FF FF FF BF :AD
7AF0 FD FF FF FF FF BF FB FF FF FF FF DF FB FF FF FF :86
-----------------------------------------------------------
Sum: 34 5D A7 B5 89 FD 68 49 87 EC E4 EC 50 2E ED F7 :C9


Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7B00 FF DF F7 FF FF FF FF EF F7 FF FF FF FF EF EF FF :90
7B10 FF FF FF F7 EF FF FF FF FF F7 DF FF FF FF FF FB :AC
7B20 DF FF FF FF FF FB BF FF FF FF FF FD BF FF FF FF :4A
7B30 FF FD 7F FF FF FF FF FE 55 55 54 15 55 55 AA AA :86
7B40 83 C2 AA AA 55 54 3F FC 15 55 AA 83 FF FF C2 AA :7E
7B50 55 3E 00 00 7C 55 AA FE 3F 1F FF 2A 54 FE 3F 1F :43
7B60 FF 55 A9 FE 00 00 FF AA 55 FE 3F 1F FF 95 AB FE :92
7B70 3F 1F FF CA 53 FE 00 00 FF D5 A7 FE 3F 1F FF EA :38
7B80 57 FE 3F 1F FF E5 AF FE 00 00 03 F2 4F FF FF FF :85
7B90 F8 F5 9F FC 63 18 FC 7A 5F F0 F1 8C 7C 79 BF E3 :DC
7BA0 F1 CE 38 FC 3E 0F F8 C7 11 FD 7F FF FF FF FF FE :86
7BB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7BC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7BD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7BE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7BF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-----------------------------------------------------------
Sum: 32 0F DC 7D B0 AB 47 CE 62 7E 33 57 6D 6A FF 34 :7E


Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7C00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7C10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7C20 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF :F8
7C30 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7C40 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7C50 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7C60 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7C70 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7C80 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7C90 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7CA0 55 55 55 55 55 55 AA AA AA AA AA AA 55 55 55 55 :4E
7CB0 55 55 AA AA AA AA AA AA 55 55 55 55 55 55 AA AA :F8
```

リスト 2　ショウギSB

```
7CC0 AA AA AA AA 55 55 55 55 55 55 AA AA AA AA AA AA :A2
7CD0 55 55 55 55 55 55 AA AA AA AA AA AA 55 55 55 55 :4E
7CE0 55 55 AA AA AA AA AA AA 55 55 55 55 55 55 AA AA :F8
7CF0 AA AA AA AA 55 55 55 55 55 55 AA AA AA AA AA AA :A2
-------------------------------------------------------
Sum: A1 A1 4B 4B A1 A1 4B 4B A0 A0 4A 4A A0 A0 4A 4A :58

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7D00 55 55 55 55 55 55 AA AA AA AA AA AA 55 55 55 55 :4E
7D10 55 55 AA AA AA AA AA AA 00 00 00 00 00 00 00 00 :A6
7D20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7D30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7D40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7D50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7D60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7D70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7D80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7D90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7DA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7DB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7DC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7DD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7DE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7DF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-------------------------------------------------------
Sum: AA AA FF FF FF FF 54 54 AA AA AA AA 55 55 55 55 :F4
```

リスト 3　ショウギDT

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6000 00 55 03 03 0F 71 07 92 08 93 04 64 06 62 FA 73 :4C
6010 FE 82 FF 1A 17 91 01 93 01 17 81 01 93 80 18 81 :1B
6020 01 83 80 07 81 01 92 80 17 73 41 02 00 1A 18 74 :12
6030 00 00 18 81 01 84 80 08 74 01 84 80 18 65 01 84 :21
6040 80 08 65 01 84 80 00 07 02 74 01 72 01 00 07 18 :02
6050 83 01 71 80 00 00 4B 03 05 0D 83 0C 86 05 91 FB :7B
6060 94 FF 97 FE A1 07 1E 07 96 42 01 0C 84 01 95 03 :F7
6070 07 84 01 95 80 07 93 45 80 07 74 01 95 80 07 64 :FC
6080 01 95 80 00 0B 0C 84 01 95 01 0C 93 45 80 00 06 :B2
6090 0C 85 00 00 00 0B 0C 85 01 96 80 07 96 41 80 00 :A2
60A0 00 6B 03 05 0F 63 08 73 06 72 07 91 FF 93 F2 51 :45
60B0 FA A1 02 1E 02 81 41 01 17 71 4E 80 07 71 4E 80 :1C
60C0 02 92 4E 80 02 82 4E 80 17 82 4E 80 17 92 41 80 :85
60D0 00 1F 17 71 01 92 01 07 71 01 82 81 17 62 01 91 :C2
60E0 02 17 52 01 91 02 17 82 41 80 07 62 01 91 80 00 :D4
60F0 10 0D 81 00 00 18 81 01 83 80 0D 72 06 82 80 00 :C2
-------------------------------------------------------
Sum: B8 E1 C5 CE FD 3E D6 07 B0 E5 08 F2 67 B3 61 4E :9C

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6100 0B 0D 61 06 71 80 18 81 41 80 00 00 40 03 05 11 :23
6110 53 06 73 03 61 FD 71 FF 72 FE A1 02 A2 02 A3 04 :FB
6120 17 02 62 43 01 02 81 41 80 02 82 42 80 04 83 42 :12
6130 80 04 63 01 81 80 00 0E 04 83 42 01 04 63 43 80 :EB
6140 02 82 42 80 00 06 02 72 00 00 00 00 4A 03 07 13 :27
6150 74 04 85 06 54 F2 61 FA 81 FF 83 FA 94 F9 A1 08 :D7
6160 A2 03 0B 03 82 01 92 01 08 63 4E 80 00 16 13 91 :BC
6170 41 01 08 81 4E 80 03 81 41 80 13 93 47 80 03 93 :E1
6180 47 80 00 0C 08 73 01 92 01 08 82 01 92 80 00 06 :85
6190 18 82 00 00 00 00 4C 03 05 11 64 08 73 07 61 F2 :38
61A0 62 F3 81 FF 92 FB A1 02 A2 04 13 02 91 01 82 01 :D5
61B0 02 82 4D 80 17 83 4E 80 17 71 41 80 00 10 04 74 :8A
61C0 01 91 01 04 94 45 80 17 63 06 73 80 00 14 17 71 :FF
61D0 00 00 07 71 00 00 17 72 0D 64 81 07 72 0D 64 81 :5E
61E0 00 00 42 02 05 13 75 06 82 08 83 06 95 06 51 FA :D0
61F0 61 FA 81 FF 73 FA A1 02 15 18 91 01 72 01 02 71 :90
-------------------------------------------------------
Sum: 73 A5 0C 58 35 BB EB 65 C7 FD 8B 6B 9A BE E1 E0 :8F

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6200 01 92 80 02 91 01 72 80 18 71 41 80 00 10 0E 81 :82
6210 41 01 0E 82 01 63 80 16 82 01 63 80 00 06 02 82 :BC
6220 00 00 00 00 5E 03 05 13 65 06 85 06 91 07 53 FA :54
6230 63 FA 83 FA 72 FF A1 08 A2 03 1B 03 61 01 82 01 :9C
6240 08 61 01 82 03 03 81 01 73 80 08 81 01 73 80 17 :FB
6250 92 01 73 80 00 10 08 73 41 01 08 71 01 91 80 08 :E6
6260 93 01 91 80 00 0B 17 71 00 00 07 71 01 82 80 00 :B3
6270 11 03 73 01 90 80 03 71 01 91 80 03 93 01 91 80 :C6
6280 00 00 3D 02 03 11 53 06 73 03 51 FA 71 FF 91 FB :69
6290 92 F3 A1 08 A2 02 22 08 93 4D 01 08 82 4D 80 08 :3C
62A0 62 4D 80 02 61 41 80 02 81 41 80 02 62 4D 80 02 :CA
62B0 72 4D 80 02 82 4D 80 00 06 02 72 00 00 00 00 57 :61
62C0 03 05 13 63 06 51 FA 71 FF 81 FC 73 FA 83 FA 93 :39
62D0 FA A1 07 A2 02 14 07 61 01 82 01 02 62 01 82 03 :30
62E0 02 72 41 04 02 82 41 80 00 0F 02 72 41 01 17 62 :3C
62F0 04 72 80 02 92 41 80 00 0B 17 62 00 00 07 62 01 :39
-------------------------------------------------------
Sum: 4C 0A 42 1A 19 CD 72 69 EE 49 80 5A 7A CA 7C F2 :36

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6300 71 80 00 0B 07 72 01 91 80 02 72 41 80 00 07 07 :CA
6310 62 01 71 80 00 00 4F 02 03 0F 85 06 81 07 71 F8 :33
6320 91 FB 93 FF 63 F8 A1 03 14 03 94 41 01 17 84 01 :A6
6330 92 02 03 84 01 94 80 03 82 48 80 00 0B 17 84 00 :23
6340 00 17 91 01 85 80 00 07 03 83 01 81 01 00 16 03 :D7
6350 74 06 82 80 13 74 06 82 80 03 94 06 82 80 03 94 :41
6360 06 82 80 00 00 4B 03 07 17 84 04 95 03 71 F2 74 :6B
6370 FA 83 FA 91 F8 92 FB 93 FF A1 02 A2 04 A3 04 0C :1B
6380 04 85 01 82 01 02 94 01 82 80 00 10 04 94 01 81 :D0
6390 01 02 73 01 81 80 14 92 41 80 00 0E 04 93 45 01 :CA
63A0 02 72 4E 80 04 73 48 80 00 06 02 92 00 00 00 00 :1B
63B0 58 04 05 13 44 05 42 07 62 07 74 06 43 F8 53 FA :71
63C0 51 FF 63 F8 A1 02 21 17 41 48 01 07 41 48 01 17 :B8
63D0 61 01 42 80 07 61 01 42 80 02 41 48 80 02 61 48 :05
63E0 80 02 52 C8 63 80 00 10 02 61 48 01 02 42 01 62 :E2
63F0 80 02 41 01 62 80 00 10 17 42 00 00 15 41 01 62 :C8
-------------------------------------------------------
Sum: 7B A1 93 77 32 2C C9 4F B1 01 A6 4C BA B5 8C B6 :F1

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6400 80 15 42 01 62 80 00 00 53 02 03 0F 82 07 93 08 :45
6410 52 FA 62 FA 71 FF 72 FA 75 F8 32 17 84 01 61 01 :21
6420 07 84 01 61 01 07 81 41 80 17 81 41 80 17 83 08 :32
6430 93 80 07 83 08 93 80 07 85 08 93 80 17 85 08 93 :96
6440 80 17 92 08 93 80 07 92 08 93 80 00 0E 07 81 00 :8E
6450 00 17 81 00 00 0D 81 00 00 00 00 3A 02 05 13 74 :EE
6460 08 83 04 72 F3 81 FF 82 FD 84 FA 94 FA A1 02 A2 :44
6470 02 12 02 92 41 01 02 91 43 80 02 71 43 80 14 91 :1B
6480 43 80 00 0B 14 91 41 01 02 91 01 93 80 00 06 02 :64
6490 92 00 00 00 00 44 02 03 13 53 08 63 07 52 FD 81 :83
64A0 FC 82 FF 83 FA 91 FB 93 FA A1 07 18 07 72 41 01 :8E
64B0 17 62 06 72 80 18 71 41 80 08 71 41 80 27 62 06 :84
64C0 72 02 00 0B 17 62 00 00 07 62 01 73 80 00 0A 17 :76
64D0 72 41 81 07 72 41 81 00 00 5B 03 05 17 52 06 64 :A5
64E0 04 93 04 51 FA 74 FD 83 FA 85 FD 92 FF A1 07 A2 :31
64F0 07 A3 08 15 07 91 41 01 07 72 04 82 80 08 81 01 :AA
-------------------------------------------------------
Sum: CD B3 57 63 BB 4E 6A 43 AC F1 4B 01 0E B7 67 F3 :F8

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6500 93 80 07 62 04 82 80 00 16 08 73 01 92 01 07 81 :2F
6510 01 92 80 07 71 01 92 80 07 61 01 92 80 00 15 07 :35
6520 91 00 00 07 82 01 93 80 07 72 01 93 80 07 62 01 :25
6530 93 80 00 00 3D 01 01 11 83 06 84 04 96 08 97 05 :AE
6540 42 F3 53 F3 81 FA 91 FF 28 18 52 00 00 08 52 00 :72
6550 00 18 63 06 92 80 08 63 06 92 80 18 41 06 92 80 :87
6560 08 41 06 92 80 08 74 06 92 80 08 85 06 92 80 00 :9A
6570 00 35 03 03 15 64 0D 73 02 86 06 96 06 63 F9 81 :3B
6580 FE 83 FA 93 FF 94 FA A1 02 16 0D 84 46 01 02 84 :B2
6590 46 80 02 83 47 80 02 92 42 80 0D 94 41 80 00 06 :D0
65A0 22 83 00 00 00 00 82 05 05 1B 34 06 63 04 64 06 :57
65B0 74 07 85 06 44 FA 53 FD 54 FA 62 FF 83 FA 92 FE :50
65C0 94 FA A1 08 19 08 73 01 72 01 17 71 01 52 03 08 :25
65D0 51 01 52 05 17 72 41 80 17 73 41 80 00 2A 18 84 :04
65E0 01 82 01 08 84 01 82 01 08 51 01 82 80 18 51 01 :5A
65F0 82 80 18 62 41 80 08 62 41 80 18 82 41 80 08 82 :4D
-------------------------------------------------------
Sum: 44 9D D3 91 5B 74 CF 05 D8 81 FA 6F A4 A6 DE 2C :FE

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6600 41 80 14 71 41 80 00 06 17 71 00 00 00 0C 0D 51 :FF
6610 01 43 01 08 61 01 42 80 00 07 08 52 01 32 80 00 :85
6620 07 17 72 06 62 80 00 00 35 02 03 0B 73 0E 82 FE :BE
6630 92 FF A1 07 A2 03 1B 03 83 42 01 03 81 41 80 03 :0A
6640 93 41 80 07 91 41 80 07 93 42 80 07 94 06 93 80 :BD
6650 00 0B 07 91 00 00 07 72 01 93 80 00 00 4A 02 03 :7F
6660 0D 73 0E 41 FA 71 FF 82 F2 A1 07 A2 03 1C 07 91 :AE
6670 4E 01 07 51 06 61 02 07 81 41 80 07 61 41 80 03 :85
6680 72 4E 80 03 62 01 81 80 00 0B 03 72 00 00 03 62 :8C
6690 01 81 80 00 0B 03 62 01 81 01 17 61 41 80 00 07 :35
66A0 17 61 01 92 80 00 00 49 02 03 13 54 06 71 07 81 :3F
66B0 08 53 F9 61 FE 82 FC 91 FF 92 FA A1 04 22 08 63 :7F
66C0 02 71 01 18 63 02 71 01 08 72 02 71 02 18 72 02 :DE
66D0 71 02 08 92 41 80 18 92 41 80 04 83 47 80 00 0A :91
66E0 04 83 00 00 18 81 41 80 00 06 04 83 47 80 00 00 :35
66F0 50 02 03 0F 96 08 97 05 81 FC 82 FD 91 FF 98 F9 :BB
-------------------------------------------------------
Sum: 22 14 CA 5F 74 A8 25 FE 22 08 46 4C 59 64 C7 BB :99

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6700 A1 07 28 07 92 41 01 07 93 43 80 07 94 06 93 80 :BC
6710 07 95 06 93 80 08 74 07 97 80 08 85 17 97 80 18 :22
6720 63 17 97 80 08 63 17 97 80 00 15 08 74 00 00 08 :C3
6730 85 07 97 80 08 63 17 97 80 08 63 17 97 80 00 00 :D5
6740 5E 05 07 1D 14 0D 35 04 55 08 22 FF 11 FB 21 FC :88
6750 31 FB 13 FC 23 FC 33 FA 34 F8 53 FA A1 02 A2 04 :49
6760 16 0D 23 48 01 14 23 48 80 0D 13 45 80 18 33 45 :03
6770 80 08 33 45 80 00 0E 04 14 48 01 04 34 48 80 14 :03
6780 23 41 02 00 07 04 34 01 32 02 00 07 04 35 01 14 :2F
6790 80 00 0B 02 42 00 00 02 43 01 41 81 00 00 77 02 :50
67A0 07 1D 96 06 86 06 76 06 62 0E 91 F3 93 FF 94 FA :DC
67B0 83 FA 72 FD 73 FA 63 FA A1 03 A2 03 A3 04 13 0E :C7
67C0 71 4D 02 04 85 01 82 01 03 84 46 80 03 82 4D 80 :6C
67D0 00 0A 03 93 4D 80 03 71 4D 80 00 0F 04 85 01 82 :C9
67E0 02 03 84 46 80 03 82 4D 80 00 0A 03 71 4D 02 03 :71
67F0 93 4D 80 00 0F 03 93 01 81 02 14 73 43 80 0E 73 :54
-------------------------------------------------------
Sum: E8 CE EA 22 7D B7 E3 49 10 3A 61 70 11 86 06 8F :69

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6800 43 80 00 0B 03 93 01 91 80 03 91 4D 80 00 06 23 :00
6810 92 00 00 00 00 58 03 07 15 84 05 72 07 73 06 91 :15
6820 FF 81 FE 82 FA 63 FD A1 02 A2 02 A3 04 13 02 92 :EF
```

リスト 3　ショウギDT

```
6830 42 01 15 82 42 80 17 82 42 80 04 93 01 92 80 00 :A1
6840 17 02 81 41 01 04 83 42 80 15 82 42 80 17 82 42 :59
6850 80 17 71 02 81 80 00 0B 04 93 01 91 01 15 82 42 :19
6860 80 00 0A 17 71 00 00 07 71 00 00 00 00 5B 03 05 :ED
6870 15 73 03 82 07 63 F2 74 F8 81 FC 91 FF 94 FA 84 :F4
6880 FA A1 02 A2 05 1E 05 93 44 01 05 92 48 80 17 81 :36
6890 4E 80 07 81 4E 80 07 92 41 80 17 92 41 80 02 92 :7C
68A0 48 80 00 1E 07 92 41 01 17 92 41 01 02 92 48 80 :08
68B0 07 81 4E 80 17 81 4E 80 02 92 48 80 07 81 4E 80 :6E
68C0 00 06 02 82 00 00 00 00 4E 03 03 17 65 06 75 06 :DB
68D0 85 06 93 06 64 04 71 07 82 FF 63 FA 83 FA 92 FA :EB
68E0 A1 02 21 17 73 41 02 07 72 01 93 80 17 72 01 93 :3B
68F0 01 07 81 41 01 17 81 41 01 14 72 01 93 80 16 92 :E7
------------------------------------------------------
Sum: 00 C5 A0 8C 82 C2 1C 78 A7 8E 2B 90 30 38 5C 8B :08


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6900 41 80 00 07 03 72 01 91 80 00 0B 02 72 00 00 02 :D0
6910 74 01 62 80 00 00 6D 03 05 11 71 03 61 0D 65 06 :2A
6920 81 FF 94 FD 72 FE 73 FA A1 08 1D 0D 72 41 02 08 :7E
6930 92 01 91 01 03 82 41 80 13 82 41 80 03 62 05 71 :9C
6940 80 13 62 05 71 80 00 14 03 82 41 02 13 82 41 02 :9F
6950 03 62 06 71 80 13 62 06 71 80 00 15 08 61 41 02 :89
6960 08 81 01 83 80 02 82 01 63 80 02 62 01 83 80 00 :5D
6970 0C 0D 81 01 93 80 18 81 01 93 80 00 06 02 62 00 :C5
6980 00 00 00 51 03 05 19 51 08 61 08 65 06 85 06 83 :AD
6990 FA 92 FE 63 FA 73 FA 82 FF 93 FA A1 02 A2 04 17 :C2
69A0 04 74 46 01 08 73 41 80 18 73 41 80 02 72 01 91 :4D
69B0 80 04 94 46 80 00 17 08 73 41 01 18 73 41 01 02 :81
69C0 72 01 91 80 02 81 41 80 18 72 41 80 00 06 02 72 :8D
69D0 00 00 00 00 49 02 05 11 73 06 95 06 63 FA 81 FF :52
69E0 92 FE A1 08 A2 02 A3 03 16 03 72 01 82 01 02 72 :06
69F0 01 91 80 08 72 01 82 80 02 71 01 82 80 00 18 08 :25
------------------------------------------------------
Sum: E2 1E FB 0A 60 78 F4 19 46 44 2A B2 4C F3 79 9D :A5


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6A00 91 41 01 02 83 42 80 02 81 01 73 80 08 71 01 73 :7E
6A10 80 08 93 42 80 00 06 02 81 00 00 00 00 36 03 05 :A4
6A20 0F 62 03 72 07 91 FF 93 F2 A1 02 A2 04 A3 06 0E :02
6A30 06 92 4E 01 04 83 4E 80 02 81 41 80 00 0F 02 81 :12
6A40 4E 01 04 83 4E 80 17 71 06 81 80 00 06 04 83 00 :C0
6A50 00 00 00 5B 02 05 15 62 07 72 08 95 06 73 FA 83 :E5
6A60 FA 93 FA 74 F8 92 FF A1 04 A2 04 20 04 84 46 01 :BE
6A70 08 81 41 80 18 81 41 80 08 83 41 80 18 83 41 80 :4C
6A80 08 63 04 82 80 18 63 04 82 80 00 1C 18 83 41 01 :EB
6A90 08 83 41 01 08 81 41 80 18 81 41 80 08 63 04 82 :62
6AA0 80 18 63 04 82 80 00 06 04 75 00 00 00 00 3C 02 :BE
6AB0 05 11 63 03 43 0E 73 FD 61 FE 51 FF 41 FE A1 02 :CE
6AC0 A2 04 15 0E 33 02 42 01 0E 61 01 42 80 02 42 42 :F9
6AD0 80 02 52 C2 61 80 00 0C 04 43 01 41 01 02 52 C2 :23
6AE0 61 80 00 06 02 31 00 00 00 00 3D 02 05 11 43 07 :B9
6AF0 44 06 71 FA 61 F8 51 FF 31 F2 A1 07 A2 02 13 27 :07
------------------------------------------------------
Sum: D2 ED 07 E3 B2 C0 E9 9E 51 45 F5 FE BD D2 1C C4 :9A


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6B00 41 4E 01 02 62 41 80 02 52 48 80 27 53 06 52 80 :23
6B10 00 0F 17 53 0E 52 01 02 62 41 80 02 52 4E 80 00 :21
6B20 06 02 42 00 00 00 00 64 04 05 17 61 03 62 07 35 :D0
6B30 06 41 FF 32 FD 33 F8 22 FA 11 FB 13 FA A1 07 A2 :1F
6B40 04 1E 07 21 43 04 17 52 01 31 80 13 52 01 31 80 :C3
6B50 03 52 01 31 80 07 51 48 80 04 53 01 31 01 00 06 :B7
6B60 07 41 43 01 00 07 14 41 01 21 01 00 07 03 32 01 :48
6B70 12 80 00 11 17 52 01 31 01 13 52 01 31 80 03 52 :AB
6B80 01 31 80 00 06 04 43 00 00 00 00 61 03 09 15 84 :05
6B90 04 71 0D 91 FF 92 FD 81 F9 82 FA 51 FE A1 03 A2 :2C
6BA0 03 A3 04 0E 04 83 43 01 14 92 41 80 0D 81 43 80 :3B
6BB0 00 0A 0D 81 41 01 0D 82 47 80 00 17 03 72 43 01 :00
6BC0 03 92 01 71 80 07 91 41 80 07 61 42 80 07 71 41 :C3
6BD0 80 00 13 07 91 41 01 03 92 01 71 80 07 61 42 80 :1E
6BE0 07 71 41 80 00 06 03 92 00 00 00 00 6F 04 0B 15 :67
6BF0 72 02 73 03 92 FF 95 F4 83 FA 61 F9 A1 03 A2 05 :26
------------------------------------------------------
Sum: 71 25 0A 06 34 91 B0 64 1E 9E A6 B6 05 E8 44 B2 :7A


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6C00 A3 06 A4 06 14 05 94 4C 01 02 82 01 93 80 06 93 :7E
6C10 41 80 13 82 01 93 80 00 10 06 93 41 01 02 82 01 :DA
6C20 93 80 13 82 01 93 80 00 0C 23 82 01 92 01 03 82 :86
6C30 01 84 80 00 0E 06 93 4C 01 13 91 47 80 03 81 47 :2F
6C40 80 00 0E 13 91 41 01 13 93 41 80 03 93 41 80 00 :32
6C50 0A 13 82 00 00 02 82 00 00 00 00 59 03 07 1B 73 :14
6C60 06 54 03 33 06 61 FA 51 FC 41 FA 62 F8 52 FF 42 :66
6C70 F8 A1 02 A2 02 A3 04 A4 04 17 02 53 C8 42 01 04 :09
6C80 64 48 80 04 44 48 80 02 63 44 80 02 43 44 80 00 :6E
6C90 12 04 44 48 01 04 64 48 80 02 43 44 80 02 43 44 :65
6CA0 80 00 0B 04 64 01 42 01 02 63 44 80 00 06 02 32 :9A
6CB0 00 00 00 00 93 07 0B 23 51 08 46 06 25 06 15 06 :B3
6CC0 61 FD 62 FC 63 FA 52 FF 54 FA 44 FA 32 FB 21 FC :40
6CD0 11 FB 13 FA A1 08 A2 03 A3 04 14 03 53 01 43 01 :BD
6CE0 08 34 45 80 08 41 01 51 80 04 64 46 80 00 0F 04 :5D
6CF0 35 45 02 18 42 01 34 01 08 34 41 80 00 0C 0E 24 :47
------------------------------------------------------
Sum: A5 4F 6A D0 47 10 02 62 66 BE EE 2A E9 BC 02 B7 :83


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6D00 01 43 80 04 26 01 35 80 00 10 18 42 01 34 02 08 :4D
6D10 34 41 01 13 42 01 53 80 00 07 0E 24 01 43 80 00 :9C
6D20 0B 08 43 01 23 01 08 23 41 80 00 0C 18 32 01 12 :D0
6D30 80 0E 32 01 12 01 00 0F 08 34 00 00 18 34 00 00 :6B
6D40 0E 21 01 23 81 00 00 41 02 05 0F 62 0E 32 07 51 :25
6D50 FE 41 FF 42 FD 21 FD A1 02 16 02 31 43 01 0E 51 :2A
6D60 41 80 02 52 42 80 07 42 42 80 17 42 42 80 00 12 :0F
6D70 0E 51 41 01 17 31 41 80 07 31 41 80 17 52 42 80 :CE
6D80 00 06 02 52 00 00 00 00 8D 04 07 19 85 06 64 06 :00
6D90 91 FB 81 FC 93 FA 83 FA 72 FF 54 FA A1 08 A2 02 :1F
6DA0 A3 03 A4 03 11 03 63 01 73 03 02 63 01 82 02 08 :2D
6DB0 63 01 73 01 00 0C 02 74 01 62 80 03 74 01 64 80 :99
6DC0 00 14 03 71 01 92 80 08 71 01 92 80 03 73 44 80 :61
6DD0 08 73 44 80 00 1A 08 82 41 02 02 74 01 82 01 23 :43
6DE0 74 01 64 80 08 62 01 64 80 08 51 01 64 80 00 14 :FA
6DF0 03 71 01 92 80 08 71 01 92 80 08 73 44 80 08 73 :CD
------------------------------------------------------
Sum: 31 CB 7F 26 A1 F5 B7 34 CD 8A 59 A8 23 68 93 08 :A0


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6E00 44 80 00 0B 03 71 41 01 02 72 01 92 80 00 06 02 :14
6E10 72 00 00 00 00 83 04 0B 1B 62 0E 53 0D 43 0D 34 :73
6E20 06 31 FC 41 FE 21 F8 22 FF 23 FA 12 FB 13 FA 51 :34
6E30 FE 42 FA 14 0D 33 01 11 07 0D 23 44 80 16 33 01 :E5
6E40 11 01 0D 32 48 04 00 06 0C 22 41 01 00 07 0D 33 :5A
6E50 01 11 01 00 06 0D 22 41 80 00 07 0D 33 01 11 01 :63
6E60 00 06 0D 22 41 01 00 07 0E 44 01 21 80 00 06 0D :85
6E70 22 41 01 00 0C 0D 33 01 11 01 16 33 01 11 80 00 :9E
6E80 06 0D 22 41 01 00 07 0E 44 01 32 01 00 0A 0E 33 :4F
6E90 00 00 16 33 00 00 00 00 25 02 03 11 31 08 24 06 :E7
6EA0 21 FB 22 FD 12 FF 14 FA A1 02 A2 05 0A 02 23 43 :16
6EB0 01 18 22 45 80 00 06 05 13 00 00 00 00 5D 03 07 :85
6EC0 0F 61 07 51 08 15 06 32 FD 23 FA 13 FF A1 07 11 :02
6ED0 07 14 01 22 01 06 14 01 22 80 07 11 01 22 80 00 :B7
6EE0 0A 17 11 41 01 18 33 41 80 00 20 18 33 01 12 01 :FF
6EF0 08 33 01 12 01 18 42 04 21 80 08 42 04 21 80 08 :45
------------------------------------------------------
Sum: 3E 2B A8 30 47 B1 43 13 AB 93 8B 32 2E DB 55 66 :4E


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6F00 24 04 21 80 08 24 04 21 80 00 0F 17 11 00 00 07 :D8
6F10 11 00 00 18 11 01 13 80 00 00 66 04 07 1B 71 07 :D2
6F20 54 FA 43 FA 33 FA 23 FA 14 FA 32 FF 21 FC 11 FB :3D
6F30 74 F9 A1 08 A2 02 A3 04 1C 04 24 46 02 02 22 41 :52
6F40 01 08 41 01 22 80 04 44 46 80 02 31 01 22 80 17 :E8
6F50 72 47 80 00 07 08 31 01 32 80 00 07 08 41 01 42 :BF
6F60 01 00 10 17 51 41 02 02 32 01 53 80 02 52 01 31 :4A
6F70 01 00 07 18 23 07 71 80 00 06 02 52 00 00 00 00 :95
6F80 35 02 11 33 0E 21 FC 22 FE 11 FF 14 FA A1 02 12 :8C
6F90 A2 04 A3 05 0F 05 13 44 01 04 23 01 12 80 02 12 :88
6FA0 41 80 00 0B 04 23 01 12 01 02 12 41 80 00 06 02 :E4
6FB0 11 00 00 00 00 4B 03 05 11 85 05 53 03 72 F9 61 :21
6FC0 F8 51 FF A1 02 A2 03 A3 04 1B 23 52 47 01 04 63 :76
6FD0 01 41 80 04 43 48 80 02 52 47 80 23 42 47 80 23 :3B
6FE0 62 47 80 00 0C 04 43 01 41 01 04 63 01 41 80 00 :E8
6FF0 0F 02 31 00 00 02 51 01 32 80 02 42 47 80 00 00 :53
------------------------------------------------------
Sum: 05 A7 B5 90 22 62 D4 64 58 71 16 18 C0 C3 CC D1 :C4


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7000 8B 07 0F 1B 73 03 75 06 91 FB 94 FA 84 FD 61 FF :A8
7010 41 FB 42 F3 34 FA A1 08 A2 02 A3 04 A4 04 10 08 :53
7020 43 01 71 01 04 53 4D 80 08 83 01 52 80 00 11 04 :4D
7030 83 01 81 01 04 63 01 81 80 02 61 01 81 80 00 15 :E9
7040 08 54 01 92 01 18 54 01 92 01 14 91 41 80 02 71 :C9
7050 01 92 80 00 0B 14 91 41 01 02 82 01 93 80 00 0B :A8
7060 05 93 43 01 04 83 01 92 80 00 0C 04 83 01 92 01 :9D
7070 02 81 01 92 80 00 0E 0E 81 41 01 08 81 41 01 18 :58
7080 81 41 01 00 06 02 91 00 00 00 00 42 03 05 0D 92 :45
7090 07 74 08 91 FE 81 FF 83 FD A1 03 1C 03 72 43 01 :8B
70A0 07 91 41 80 17 91 41 80 08 63 01 92 80 18 63 01 :BC
70B0 92 80 03 82 42 80 00 0F 17 91 41 01 07 91 01 82 :6D
70C0 80 17 72 41 80 00 06 02 92 00 00 00 00 57 03 05 :C3
70D0 13 71 08 84 06 63 06 91 FD 94 FA 81 FF 73 FA A1 :29
70E0 08 A2 02 12 08 92 41 04 08 72 41 01 02 82 43 80 :A0
70F0 02 92 41 80 00 07 16 62 01 81 01 00 06 02 82 43 :24
------------------------------------------------------
Sum: 60 80 12 1F 2A F2 8C FC 03 E2 BD 62 95 31 8D 34 :40


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7100 01 00 07 03 83 01 93 80 00 14 18 93 41 01 02 83 :28
7110 01 81 80 02 93 01 81 80 18 82 43 80 00 06 22 83 :A1
7120 00 00 00 00 94 05 09 19 93 03 95 06 84 08 85 0D :0A
7130 91 FB 83 FF 73 FA 62 FE 63 FA 53 FA 65 F2 A1 03 :80
7140 1D 13 82 41 03 18 73 41 01 08 73 41 01 08 75 01 :FE
7150 72 02 03 92 01 72 80 0D 94 01 72 80 00 11 03 82 :26
7160 01 64 80 0D 84 01 72 80 06 74 01 72 80 00 0B 0D :EE
7170 82 01 61 80 0D 81 41 80 00 15 18 93 01 72 01 0D :EF
7180 93 01 72 80 08 75 01 72 80 03 83 41 80 00 13 0D :5D
7190 81 41 01 0D 82 01 61 80 03 61 41 80 0E 83 4E 80 :B8
71A0 00 0B 03 82 01 72 01 03 92 45 80 00 0B 13 71 00 :ED
71B0 00 0E 94 06 83 80 00 00 B5 07 0D 23 55 08 25 03 :1C
71C0 52 FA 53 FC 43 FA 33 FA 23 FA 11 FF 12 FB 13 FA :4C
71D0 A1 02 A2 03 A3 04 A4 04 A5 04 A6 06 A7 06 0A 03 :A6
71E0 22 41 01 02 21 41 80 00 0F 04 34 01 31 01 04 14 :DA
71F0 46 80 18 33 41 80 00 14 06 32 41 03 02 42 01 21 :C8
------------------------------------------------------
Sum: 14 0E 88 AD 08 34 DF 6C 50 09 BE C6 86 6E E7 70 :06
```

リスト 3　ショウギDT

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7200 01 02 22 01 41 02 08 33 41 80 00 0B 06 22 01 11 :AA
7210 80 14 22 41 80 00 0B 06 42 01 51 80 14 42 41 80 :B3
7220 00 14 04 24 46 03 02 42 01 21 01 02 22 01 41 02 :54
7230 08 33 41 80 00 0B 06 22 01 11 80 14 22 41 80 00 :B8
7240 0B 06 42 01 51 80 14 42 41 80 00 10 02 42 01 21 :B2
7250 01 02 22 01 41 80 18 33 41 80 00 0B 06 22 01 11 :38
7260 01 14 22 41 80 00 06 04 23 00 00 00 00 69 04 0B :9D
7270 19 43 06 29 05 14 04 15 04 16 04 17 04 19 05 41 :55
7280 FA 33 FA 12 F8 13 FF A1 03 16 04 25 01 14 01 03 :3F
7290 22 01 14 80 03 24 01 14 80 15 23 01 14 80 00 0A :4A
72A0 14 13 41 01 03 23 48 80 00 0B 14 23 48 01 03 24 :09
72B0 01 14 80 00 10 03 24 01 14 01 04 24 06 14 80 15 :B9
72C0 23 41 80 00 0B 03 23 01 15 01 13 13 41 80 00 06 :19
72D0 08 24 00 00 00 00 77 04 07 1D 55 06 45 06 35 06 :AC
72E0 12 08 13 06 16 05 15 05 52 F3 53 FA 31 FA 33 FF :57
72F0 25 FA A1 08 A2 02 20 08 24 01 32 01 02 23 01 43 :55
--------------------------------------------------------
Sum: 42 7E 18 F3 EF 8B 8C 73 57 12 02 54 86 D8 FB A5 :01


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7300 80 02 44 01 42 80 02 34 01 42 80 08 44 01 42 80 :91
7310 06 34 01 42 80 00 18 18 23 41 01 08 23 41 01 02 :01
7320 23 01 41 80 02 33 01 41 80 02 22 41 80 00 0C 02 :CF
7330 33 01 24 02 02 34 01 32 01 00 0C 02 33 01 41 80 :C7
7340 18 33 01 41 80 00 06 02 34 00 00 00 00 8B 06 0D :E7
7350 1B 51 0E 52 02 44 06 42 FA 31 FC 32 FF 33 FA 21 :00
7360 FC 23 FA 11 FB 13 FA A1 08 A2 02 11 0E 42 01 22 :03
7370 02 0E 41 01 22 01 02 42 01 22 80 00 0F 02 32 01 :A0
7380 12 80 0E 31 41 80 04 34 46 80 00 0A 0E 31 41 01 :1B
7390 0E 32 41 80 00 10 06 32 41 01 22 42 01 22 80 08 :9A
73A0 42 01 22 80 00 15 04 24 46 01 08 41 01 22 80 08 :5D
73B0 43 01 22 80 02 43 01 22 80 00 0C 08 41 01 22 01 :47
73C0 02 42 01 23 80 00 07 02 23 01 31 01 00 0A 02 32 :85
73D0 00 00 18 32 00 00 00 00 41 03 07 15 34 02 31 FE :0F
73E0 33 FA 23 FA 11 F8 13 FF 14 FB A1 02 A2 04 A3 04 :64
73F0 0B 04 25 01 12 01 22 24 46 80 00 0B 04 24 46 01 :CE
--------------------------------------------------------
Sum: F2 E1 E8 6B 4B 20 6F B7 E7 7B 3C 4E 61 EF 42 9C :D1


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7400 02 13 01 21 80 00 0C 22 23 01 21 01 02 13 01 21 :62
7410 80 00 06 04 13 00 00 00 00 41 03 07 11 74 02 93 :02
7420 FF 94 FA 82 FB 71 FB A1 02 A2 04 A3 04 CB 04 85 :FA
7430 01 92 01 22 84 45 80 00 0B 04 84 45 01 02 93 01 :6E
7440 91 80 00 10 14 93 41 01 02 93 01 91 80 22 83 01 :57
7450 91 80 00 06 22 83 00 00 00 00 00 00 00 00 00 00 :BC
--------------------------------------------------------
Sum: A4 39 02 DF 48 CC C8 C4 32 7B AD 81 98 B6 1D 3B :DF
```

RANDOM BOX

『2^N-1を求める』

●68T09S

先日，『FM-7/8活用研究』を見ていたら，2^N-1を求めるプログラムが出ていました．$N=86243$のときの計算時間が7時間とあった点に私は大変驚ろきました．

I/O誌'83年10月号のZ80 (4MHz) で3時間22分であるのに，6809がこんなに遅いなど信じられません．6809の能力をかっている私はさっそく納得のいく結果を得るためプログラムを組んでみました．その結果，一応のスピードが得られたので報告したいと思います．

工夫した点としてはアセンブラ28～2A行でのキャリーフラグの処理があります．ループ回数をあらかじめ計算（20～22行）せずに，CMPなどでループエンドを判定するとキャリーフラグが破壊されてしまいます．

コールの方法は，BASICプログラムを見ればわかると思いますが，DATA部に3バイトのNを入れSTARTからEXECすれば，［BOTOM］～START-1までに結果2^N-1がBCDで入ります．

プログラムは，リロケータブルですからSTARTを$7F00ぐらいにすればNは，19万ぐらいまで計算できると思います．

$N=86243$のときの計算時間は，2MHzで2時間16秒でした．

BASICリスト

```
1000 '
1010 '   Mersenne prime number
1020 '   culculate program
1030 '   by 68T09S
1040 '83/10/14
1050 '
1060 CLEAR30,&H12FF
1070 '
1080 PRINT"SUBROOTINE LOADED ?":A$=INPUT$(1)
1090 IF A$="Y" THEN 1110
1100 LOADM"MPRIME.O"
1110 INPUT"N=";N
1120 TIME$="00:00:00"
1130 T=N:EE=N
1140 START=&H6E00
1150 TTOP=START+&H54
1160 FOR I=TTOP+2 TO TTOP STEP -1
1170 T=INT(N/256)
1180 X=N-T*256
1190 IF I=TTOP AND X>2 THEN 1240
1200 POKE I,X
1210 N=T
1220 NEXT
1230 IF T=0 THEN 1250
1240 BEEP:PRINT"OVER !!":END
1250 EXEC START
1260 TT=PEEK(TTOP+3)*256+PEEK(TTOP+4)
1270 TE=START-1
1280 PRINT"AREA ";HEX$(TT);"-";HEX$(TE):PRINT
1290 PRINT" CALCULATE TIME= ";TIME$:PRINT
1300 PRINT"     2^";EE;"-1 =":PRINT
1310 I=0
1320 FOR A=TT TO TE STEP 50
1330   PRINTUSING"######: ";I;
1340    I=I+1
1350   FOR B=A TO A+49 STEP 5
1360     FOR C=B TO B+4
1370      IF C>TE THEN 1450
1380      X=PEEK(C)
1390      OUT$=RIGHT$(STR$(X¥16),1)
1400      OUT1$=RIGHT$(STR$(X MOD 16),1)
1410      PRINTOUT$;OUT1$;
1420     NEXT
1430     PRINT;
1440    NEXT
1445   PRINT
1446 NEXT
1450 PRINT:PRINT:PRINT" WHOLE TIME= ";TIME$
1460 END
```

マシン語部分アセンブル・リスト

```
0001  0000                  ;
0002  0000                  ;Mresenne prime number
0003  0000                  ;    calculate program
0004  0000                  ;    by 68T09S
0005  0000                  ;    83/10/14
0006  0000                  ;
0007  6E00                          ORG  $6E00
0008  6E00 308CFD           START:  LEAX <START,PC
0009  6E03 8601                     LDA  #1
0010  6E05 A782                     STA  ,-U
0011  6E07 AF8C4D                   STX  <BOTOM,PC
0012  6E0A E68C47                   LDB  <DATA,PC
0013  6E0D 270C             L1:     BEQ  L2
0014  6E0F 8E0000                   LDX  #0
0015  6E12 3404                     PSHS B
0016  6E14 8D19                     BSR  ADD
0017  6E16 3504                     PULS B
0018  6E18 5A                       DECB
0019  6E19 20F2                     BRA  L1
0020  6E1B AE8C37           L2:     LDX  <DATA+1,PC
0021  6E1E 2702                     BEQ  SUB
0022  6E20 8D0D                     BSR  ADD
0023  6E22 308CDB           SUB:    LEAX <START,PC
0024  6E25 8699             L4:     LDA  #$99
0025  6E27 AB82                     ADDA ,-X
0026  6E29 19                       DAA
0027  6E2A A784                     STA  ,X
0028  6E2C 24F7                     BCC  L4
0029  6E2E 39                       RTS
0030  6E2F                  ;
0031  6E2F 338CCE           ADD:    LEAU <START,PC
0032  6E32 1F30                     TFR  U,D
0033  6E34 A38C20                   SUBD <BOTOM,PC
0034  6E37 1F02                     TFR  D,Y
0035  6E39 1CFE                     ANDC #$FE
0036  6E3B A6C4             L3:     LDA  ,-U
0037  6E3D A9C4                     ADCA ,U
0038  6E3F 19                       DAA
0039  6E40 A7C4                     STA  ,U
0040  6E42 313F                     LEAY -1,Y
0041  6E44 26F5                     BNE  L3
0042  6E46 2404                     BCC  NC
0043  6E48 8601                     LDA  #1
0044  6E4A A7C2                     STA  ,-U
0045  6E4C EF8C08           NC:     STU  <BOTOM,PC
0046  6E4F 301F                     LEAX -1,X
0047  6E51 26DC                     BNE  ADD
0048  6E53 39                       RTS
0049  6E54                  ;
0050  6E54 000000           DATA:   FCB  0,0,0
0051  6E57 0000             BOTOM:  FCB  0,0
0052  6E59                  ;
```

FM-7 迫るUFO！君はアストロノーツだ!!

3D UFO

■高畑英樹

パレット機能をフルに活用した，3次元表示の高速ゲームができましたので発表します．

プログラムについて

3次元表示で最大の問題は，画面の奥側と手前側のパターンが重なったときの処理と，さらには1番手前にきた最大パターンを徐々に画面から消す処理が高速性を要求されることです．しかし，パレットの定義を次のように決めることでハード的に解決ができるのです．

❶UFOのパターン・データは青のVRAMだけに書き込み，青のパレットを利用し，マゼンダ，水色，黄色，白色と4色の表現をしています．

❷UFOの消去は実際にソフトでは消去していません．画面の周囲を見ると緑と黄色でフレームが表示されていますが，UFOがこのエリアにきたとき，UFOの青のVRAMデータとの三原色の論理演算により見掛け上消去しています．具体的には青と緑で水色になるため水色のパレットを緑にし，また黄と青で白になるため，白のパレットを黄色にしています．

❸UFOからの攻撃は赤のパレットを利用して，20msecの速さで赤と黄に変換しています．データは赤のVRAMにのみ，書いています．

入力について

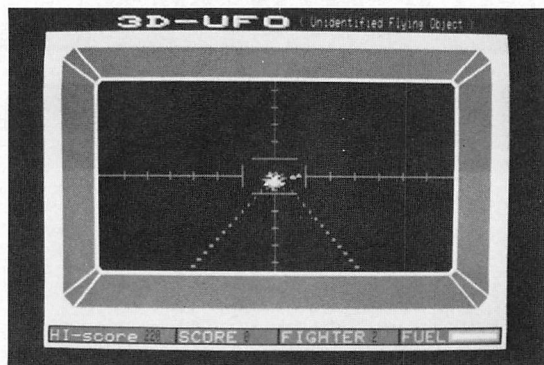
リスト1，2のBASICプログラムとマシン語リストを，

```
SAVE "3D-UFO" ⏎
SAVEM "3DUFOM", &H5000, &H5EFF, &H5000 ⏎
```

として，走らせる前にテープに取ってください．

ゲームをするときは，両方のプログラムを読み込んで，BASICからRUNで走ります．

キー操作は，タイトル画面にでるので，そちらを見てください．

せまるUFO！目標LOCK ON！

ゲームについて

UFOは4種類それぞれ異なる攻撃性を持たせてあります．白が一番活発で次に水色，マゼンダ，黄色と続きます．これはUFOの色によって攻撃したときの得点が違うことにもなります．

また，UFO自らは体当たりはしてきません．UFOを攻撃するときは，テン・キーでUFOを追跡し，ファインダーの中央へ持ってくると〝LOCK〟の表示がでます．このとき，すかさずスペース・キーでレーザービームを発射してください．LOCKの表示のないときはビームは出ません．

また，ビームの照準点は最小パターンから2～3段階の大きさで設定してあります．4段階以上近付くとLOCK不能でビームは発射できません．いっぽう，UFO側は色に応じて大きさが，4～6段階以上になると攻撃を開始します．ただし画面の中央付近にいるUFOに限ります．

参考文献

〝FM-7/8活用研究〟，工学社

リスト1 BASICプログラム・リスト

```
10 COLOR4,0:WIDTH40,20
20 HIS=0
30 GOSUB620:SYMBOL(400,140),"DEMONSTRATION",1,1,
6
40 POKE&H5E78,0
50 EXEC&H5000
60 I=PEEK(&H6FF0):IFI=(&HFF)THEN70ELSE60
70 FORI=0TO7:COLOR=(I,I):NEXT
80 CLS
90 SYMBOL(220,30),"GAME message",2,2,4
100 SYMBOL(135,70),"UFO point",2,1,2
110 SYMBOL(200,85),"=100",2,1,7
120 SYMBOL(200,101),"=50",2,1,7
130 SYMBOL(200,117),"=20",2,1,7
140 SYMBOL(200,133),"=10",2,1,7
150 POKE&H5E78,3:EXEC&H5000
160 SYMBOL(380,60),"PLAY key",2,1,2
170 SYMBOL(380,75),"UP-----5",2,1,7
```

リスト1　BASICプログラム・リスト

```
180 SYMBOL(380,91),"DOWN---2",2,1,7
190 SYMBOL(380,107),"RIGHT--3",2,1,7
200 SYMBOL(380,123),"LEFT---1",2,1,7
210 SYMBOL(380,139),"STOP---space",2,1,7
220 SYMBOL(380,155),"ATTACK-space",2,1,7
230 I$=INKEY$:I$=INPUT$(1)
240 SC=0
250 GOSUB620
260 POKE&H5E78,1
270 EXEC&H5000
280 I=PEEK(&H6FF0):IFI=(&H80)THEN320
290 IFI=(&H81)THEN510
300 I=PEEK(&H6FF1):IFI=1THENGOSUB610
310 I=PEEK(&H6FF2):IFI=2THENGOSUB600:GOTO280ELSE
280
320 SOUND6,&H1F:SOUND7,7:SOUND8,&H10:SOUND9,&H10
:SOUND10,&H10:SOUND11,0:SOUND12,25:SOUND13,0
330 COLOR=(1,0):COLOR=(2,2):X=PEEK(&H6FFD)*8+15:
Y=PEEK(&H6FFC)+6
340 LINE(X,Y)-(295,90),PSET:LINE(X,Y)-(295,105),
PSET:LINE(X,Y)-(245,75),PSET:LINE(X,Y)-(395,135)
,PSET:LINE(X,Y)-(95,60),PSET:LINE(X,Y)-(345,45),
PSET:LINE(X,Y)-(195,150),PSET:LINE(X,Y)-(495,60)
,PSET:LINE(X,Y)-(90,120),PSET:LINE(X,Y)-(545,150
),PSET
350 CIRCLE(X,Y),15,6
360 PAINT(X,Y),0,6
370 SOUND6,&H14:SOUND7,7:SOUND8,&H10:SOUND9,&H10
:SOUND10,&H10:SOUND11,0:SOUND12,50:SOUND13,0
380 FORI=1TO1000:NEXT
390 COLOR=(2,0)
400 FORI=1TO30:FORJ=3TO7:COLOR=(J,0):NEXTJ:FORK=
3TO7:COLOR=(K,2):NEXTK:NEXTI
410 COLOR=(3,2):COLOR=(4,1):COLOR=(5,1):COLOR=(6
,4):COLOR=(7,4)
420 FORI=1TO2500:NEXT
430 CIRCLE(X,Y),15,0
440 PAINT(90,45),1,6,7
450 PAINT(90,45),0,6,7
460 PL=PL-1
470 IFPL=0THEN510
480 GOSUB920
490 POKE&H5E78,2
500 GOTO270
510 SC1=PEEK(&H6FFB)*10
520 SC2=PEEK(&H6FFA)*100
530 SC3=PEEK(&H6FF9)*1000
540 SC4=PEEK(&H6FF8)*10000
550 SC=SC1+SC2+SC3+SC4
560 IFSC>HIS THENHIS=SC
570 GOSUB920
580 SYMBOL(180,70),"GAME OVER",4,2,3
590 FORI=1TO8000:NEXT:GOTO30
600 SOUND6,10:SOUND7,&H30:SOUND8,&H10:SOUND9,&H1
0:SOUND10,&H10:SOUND11,0:SOUND12,5:SOUND13,0:FOR
I=0TO250STEP7:SOUND0,I:SOUND2,I:SOUND4,I:NEXT:PO
KE&H6FF2,0:RETURN
610 SOUND6,5:SOUND7,7:FORI=1TO16:SOUND8,I:FORJ=0
TO50:NEXTJ:NEXTI:POKE&H6FF1,0:RETURN
620 CLS'----TVSCREEN
630 COLOR=(0,0):COLOR=(3,2):COLOR=(4,1):COLOR=(5
,1):COLOR=(6,4):COLOR=(7,4)
640 SYMBOL(101,1),"3D-UFO",5,1,3
650 SYMBOL(103,1),"3D-UFO",5,1,3
660 SYMBOL(100,0),"3D-UFO",5,1,6:SYMBOL(350,0),"
( Unidentified Flying Object )",1,1,3
670 LINE(0,10)-(639,199),PSET,6,B'----1
680 CONNECT(27,28)-(44,21)-(595,21)-(612,28)-(61
2,167)-(595,174)-(44,174)-(27,167)-(27,28),4'---
-2
690 CONNECT(75,151)-(75,43)-(84,40)-(555,40)-(56
4,43)-(564,151)-(555,154)-(84,154)-(75,151),6'--
--3
700 CONNECT(79,151)-(79,43)-(85,41)-(554,41)-(56
0,43)-(560,151)-(554,153)-(85,153)-(79,151),6'--
--4
710 PAINT(77,100),6:PAINT(561,100),6
720 PAINT(320,11),6,4
730 PAINT(320,22),4,6
740 CONNECT(27,167)-(77,151)-(84,153)-(44,174),6
'----5
750 CONNECT(44,21)-(85,40)-(77,43)-(27,28),6'---
-5
760 CONNECT(612,28)-(562,43)-(554,40)-(595,21),6
'----5
770 CONNECT(595,174)-(554,154)-(562,151)-(612,16
7),6'----5
780 LINE(6,186)-(632,196),PSET,0,BF
790 LINE(9,187)-(629,195),PSET,4,BF
800 SYMBOL(8,188),"HI-score",2,1,3
810 SYMBOL(192,188),"SCORE",2,1,3
820 SYMBOL(328,188),"FIGHTER",2,1,3
830 SYMBOL(488,188),"FUEL",2,1,3
840 LINE(188,186)-(190,196),PSET,0,BF
850 LINE(324,186)-(326,196),PSET,0,BF
860 LINE(484,186)-(486,196),PSET,0,BF
870 LINE(555,188)-(628,194),PSET,3,BF
880 LINE(560,190)-(623,192),PSET,6,BF
890 PL=3
900 I$=STR$(SC):SYMBOL(272,188),I$,1,1,0
910 I$=STR$(HIS):SYMBOL(136,188),I$,1,1,0
920 '----ﾌｧｲﾝﾀﾞｰ
930 LINE(447,188)-(457,194),PSET,4,BF
940 I$=STR$(PL):SYMBOL(440,188),I$,1,1,0
950 LINE(80,97)-(279,97),PSET,4
960 LINE(360,97)-(559,97),PSET,4
970 LINE(320,42)-(320,87),PSET,4
980 LINE(320,107)-(320,152),PSET,4
990 LINE(278,91)-(279,103),PSET,4,BF
1000 LINE(360,91)-(361,103),PSET,4,BF
1010 LINE(291,87)-(349,87),PSET,4
1020 LINE(291,107)-(349,107),PSET,4
1030 FORX=117TO270STEP32:LINE(X,95)-(X+1,99),PSE
T,4,BF:NEXT
1040 FORX=360TO550STEP32:LINE(X,95)-(X+1,99),PSE
T,4,BF:NEXT
1050 FORY=47TO80STEP10:LINE(316,Y)-(324,Y),PSET,
4:NEXT
1060 FORY=107TO150STEP10:LINE(316,Y)-(324,Y),PSE
T,4:NEXT
1070 FORI=(&H6FF0)TO(&H6FFF):POKEI,0:NEXT
1080 RETURN
```

リスト2　マシン語ダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5000 B6 FD 05 2B FB 1A 50 86 80 B7 FD 05 B7 FD 05 CE :8E
5010 FC 80 30 8D 00 98 C6 4A A6 80 A7 C0 5A 26 F9 7F :66
5020 FD 05 7F FD 05 8E 00 C8 30 1F 26 FC CE 5D 08 8E :0B
5030 3E 80 86 08 C6 30 8D 33 7F FD 05 7F FD 05 CE 5A :2C
5040 F0 8E D0 60 86 0C C6 32 8D 21 7F FD 05 7F FD 05 :E8
5050 33 8D 00 FB 8E BE 80 86 54 C6 34 8D 0E 86 01 B7 :34
5060 FC 80 7F FD 05 7F FD 05 16 00 8B FD 6F FE B6 FD :3C
5070 05 2B FB 86 80 B7 FD 05 B7 FD 05 BF FC FE 3A 10 :A6
5080 8E FC CA A6 C0 A7 A0 5A 26 F9 F6 6F FF F7 FC 80 :51
5090 7F FD 05 7F FD 05 B6 FD 05 2B FB 86 80 B7 FD 05 :9F
50A0 B7 FD 05 B6 FC 80 26 E8 7A 6F FE 26 C6 39 00 00 :05
50B0 3F 59 41 4D 41 55 43 48 49 93 D3 8F 90 B7 D4 0A :AA
50C0 B6 D4 08 B6 D4 09 B6 D4 0A 7F D3 80 7F D3 80 7F :DC
50D0 D3 80 B6 D4 0A B6 D3 80 27 F8 81 01 27 15 B7 D4 :58
50E0 0A B7 D4 0A FE D3 FE 8E D3 CA E6 80 E7 C0 4A 26 :16
50F0 F9 20 D6 7E BE 9E 30 8D 00 06 BF FF F6 1C 00 39 :95
-----------------------------------------------------------
Sum: A0 42 01 D5 F3 21 59 83 75 A4 CD 30 B2 E8 10 3F :A7


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5100 34 57 F6 FD 04 C4 02 26 09 8E 01 E0 BF FF F6 35 :CF
5110 57 3B C6 80 F7 FD 05 F7 FD 05 F7 FD 05 8E FC F0 :3D
5120 CE 6F F0 E6 84 27 FC C6 10 A6 80 A7 C0 5A 26 F9 :96
5130 B6 6F F0 2A 08 7F FD 05 7F FD 05 20 CC 7F FC F0 :A0
5140 FC 6F FE FD FD 39 35 57 7F FD 05 7F FD 05 3B C6 :2B
5150 D0 1F 9B 8E C0 00 6F 80 8C CF A0 26 F9 B7 D3 F0 :5B
5160 B7 D3 F0 B7 D3 F0 B7 D3 F0 B6 D4 04 39 B7 D4 0A :CA
5170 B7 D4 0A 86 7F 1F 8B 00 00 00 00 00 00 00 00 16 :5A
5180 03 25 A6 23 27 16 4A 27 25 4A 27 3E 4A 27 5E 4A :8C
5190 10 27 00 9E 4A 10 27 02 29 16 02 A0 CC 00 00 ED :F2
51A0 84 17 00 F4 96 20 27 1A CC 03 C0 ED 84 39 8D 09 :55
51B0 17 00 E5 96 20 27 2F 20 09 CE 00 00 EF 84 EF 88 :E9
51C0 50 39 CE D0 60 86 02 16 00 5A 8D 09 17 00 C9 96 :8B
51D0 20 27 44 20 11 34 10 86 06 C6 50 CE 00 00 EF 84 :E3
51E0 3A 4A 26 FA 35 90 CE D0 64 86 06 20 37 8D 0B 17 :FD
51F0 00 A6 96 20 10 27 00 80 20 1D 34 10 30 1F C6 50 :F9
-----------------------------------------------------------
Sum: A1 58 88 AA 73 8D 8D E1 3D AC F6 1F 86 69 59 2D :0C


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5200 CE 00 00 86 05 EF 01 3A 4A 26 FA 86 06 EF 84 EF :DB
5210 02 3A 4A 26 F8 35 90 CE D0 70 86 05 8D 06 30 1F :E4
5220 86 06 20 62 97 1E EC C1 ED 84 30 88 50 0A 1E 26 :37
5230 F5 39 8D 0B 17 00 61 96 20 10 27 01 C3 20 39 34 :7C
5240 10 30 1F 86 04 C6 50 CE 00 00 EF 01 EF 03 3A 4A :33
5250 26 F8 30 89 00 F0 6F 01 6F 04 3A 6F 01 6F 04 3A :01
5260 EF 84 EF 04 3A EF 84 EF 04 3A 86 05 EF 84 EF 02 :2F
5270 EF 04 3A 4A 26 F6 35 90 CE D0 92 86 07 8D 07 30 :D9
5280 1F 86 09 16 01 A0 97 1E EC C1 ED 84 EC C1 ED 02 :D4
5290 30 88 50 0A 1E 26 F1 39 86 01 97 20 A6 26 26 F7 :A7
52A0 A6 23 48 97 1E 86 64 90 1E 97 1E A6 23 81 06 10 :73
52B0 27 00 B4 6A 27 26 1D 86 09 A0 23 A7 27 6C 23 0F :6D
52C0 20 6A A4 30 88 B0 A6 23 81 04 27 04 81 06 26 04 :C0
52D0 6A 21 30 1F 96 FD 26 11 6A 28 26 BB 86 06 A0 23 :66
52E0 A7 28 A6 23 10 26 00 7F 39 E6 23 C1 04 24 42 4A :04
52F0 26 0D 6C 25 A6 25 84 01 26 9D 30 01 6C 21 39 4A :18
-----------------------------------------------------------
Sum: D2 1A AA 2E 47 47 AF CE 4B E0 7D 81 DF C7 BC F1 :4B


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5300 26 0C A6 23 10 26 00 A6 30 88 B0 6A A4 39 4A 26 :F6
5310 0F 6C 25 A6 25 84 01 10 26 FF 7C 30 1F 6A 21 39 :B4
5320 4A 10 26 FF 72 A6 23 10 26 00 71 30 88 50 6C A4 :79
5330 39 4A 26 0B C6 03 A6 A4 91 1E 25 41 5C 20 3E 4A :E0
5340 26 0B C6 02 A6 21 81 28 25 33 5C 20 30 4A 26 0B :E8
5350 C6 01 A6 A4 91 1E 24 25 5C 20 22 C6 01 A6 21 81 :B6
5360 28 25 1A C6 04 20 16 EC A4 C1 28 25 09 C6 03 91 :68
5370 1E 25 0A 5C 20 07 C6 01 91 1E 24 01 5C 5A 26 06 :4D
5380 30 1F 6A 21 20 16 5A 26 06 30 1F 6A 21 20 1F 5A :09
5390 26 06 30 01 6C 21 20 16 30 01 6C 21 86 50 E6 23 :BD
53A0 3D 9F 1E D3 1E 1F 01 A6 A4 AB 23 A7 A4 39 86 50 :7D
53B0 E6 23 3D DD 1E 1F 10 93 1E 1F 01 A6 A4 A0 23 A7 :F5
53C0 A4 39 8D 0B 17 FE D1 96 20 10 27 00 AF 20 31 34 :7C
53D0 10 30 1E 86 09 C6 50 CE 00 00 EF 02 EF 04 3A 4A :39
53E0 26 F8 86 07 EF 84 6F 02 6F 05 EF 06 3A 4A 26 F4 :96
53F0 86 05 EF 84 EF 02 EF 04 EF 06 3A 4A 26 F4 35 90 :3A
-----------------------------------------------------------
Sum: C3 75 BC 89 8E 78 55 83 39 ED 7A 41 2A CE F9 E6 :13


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5400 CE D0 E4 86 09 17 FE 7E 30 1E 86 0C 97 1E EC C1 :E6
5410 ED 84 EC C1 ED 02 EC C1 ED 04 EC C1 ED 06 30 88 :03
5420 50 0A 1E 26 E9 39 97 1E EC C1 ED 84 EC C1 ED 02 :2F
5430 EC C1 ED 04 30 88 50 0A 1E 26 ED 39 8D 05 17 FE :C1
5440 57 20 39 34 10 30 1E 86 06 C6 50 CE 00 00 EF 02 :A3
5450 EF 04 EF 06 3A 4A 26 F6 30 89 01 90 86 09 EF 84 :D4
5460 6F 02 6F 07 EF 08 3A 4A 26 F4 86 06 EF 84 EF 02 :6C
5470 EF 04 EF 06 EF 08 3A 4A 26 F2 35 90 CE D1 68 86 :CD
5480 0B 8D A3 30 1E 86 0F 97 1E EC C1 ED 84 EC C1 ED :8B
5490 02 EC C1 ED 04 EC C1 ED 06 EC C1 ED 08 30 88 50 :EA
54A0 0A 1E 26 E5 39 00 00 B6 3F F0 97 F1 27 47 4A 27 :B8
54B0 44 4A 27 5E 8E 1A 03 17 FD 5D 30 89 01 91 17 FD :8E
54C0 56 30 89 01 91 17 FD 4F 8E 5A 03 17 FD 49 30 89 :05
```

リスト 2　マシン語ダンプ・リスト

```
54D0 06 91 17 FD 42 30 89 01 91 17 FD 3B 8E 9A 03 17 :C9
54E0 FD 35 30 89 01 91 17 FD 2E 30 89 06 91 17 FD 27 :4A
54F0 86 FF 7E BE 80 0F 4E 0F 4F 0F 50 0F 51 86 01 97 :D9
-----------------------------------------------------------
Sum: D5 1F 60 5D 74 D7 47 24 A5 13 7A 39 61 BC 30 16 :35
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5500 F3 86 02 97 F4 8E 7B A6 9F F5 86 FF 97 FE 86 01 :EA
5510 97 FF 8E 7E 80 6F 80 8C 7F 4E 26 F9 86 14 97 FB :B5
5520 7F D3 60 86 01 97 17 96 F3 8B 04 81 04 26 01 4A :F5
5530 97 46 97 54 96 00 27 05 86 FF 7E BE 80 0C 21 96 :8E
5540 17 2A F1 86 01 97 17 8E 7F 46 CE D3 F0 A6 C4 26 :DB
5550 21 86 10 E6 84 27 1B E6 80 E7 C0 4A 26 F9 96 46 :B5
5560 2A 03 7E BE 80 B6 D4 04 8E 7F 46 86 08 6F 80 4A :91
5570 26 FB 96 23 26 0C 96 29 26 08 96 2F 26 04 96 35 :B3
5580 27 10 0C F8 96 F8 C6 02 84 01 26 02 C6 06 D7 55 :36
5590 D7 46 12 B6 D0 0E 81 0A 10 25 00 89 0F FF 8E 7E :26
55A0 80 86 0A C6 09 6D 84 10 26 00 7A 3A 4A 26 F6 7F :9F
55B0 D0 0E 96 FE 80 0A 97 FE 0A F4 26 2E 86 03 97 F4 :F7
55C0 9E F5 0C F6 26 02 0C F5 6F 84 6F 88 50 6F 89 00 :F0
55D0 A0 30 89 40 00 6F 84 6F 88 50 6F 89 00 A0 8C BB :B2
55E0 AD 26 07 86 81 97 46 16 FF 4A 96 F3 0C F3 81 03 :29
55F0 26 02 0F F3 97 1E 26 06 C6 06 86 02 20 16 4A 26 :05
-----------------------------------------------------------
Sum: 87 83 05 5D 63 B7 33 08 CA BF 58 02 06 9C 81 F1 :B8
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5600 06 C6 05 86 05 20 0D 4A 26 06 C6 07 86 01 20 04 :77
5610 C6 04 86 64 D7 F7 97 F0 96 1E 8B 04 81 04 26 01 :F8
5620 4A 97 54 97 46 96 F1 26 16 B6 D3 60 26 0C 0F F4 :F3
5630 86 46 97 FE 86 07 97 F7 20 1A 86 FF 7E BE 80 B6 :AD
5640 D3 60 84 0F 97 FD 81 05 26 02 0A FD 96 FD 81 05 :28
5650 25 02 0F FD 10 8E 7E 80 A6 A4 10 27 03 4A A6 26 :69
5660 27 15 A6 22 26 0C B6 D3 60 81 20 26 0A 6C 22 17 :95
5670 04 4D 17 01 87 20 03 17 03 89 E6 24 27 0E 17 00 :0C
5680 AA 31 29 10 8C 7E DA 26 CF 16 FE A8 A6 23 91 F7 :FA
5690 25 EF 8E 7E 86 C6 09 86 0A 97 1E A6 84 26 E2 3A :26
56A0 0A 1E 26 F7 EC A4 8B 09 CB 02 81 5B 24 41 81 40 :38
56B0 25 CF C1 27 24 22 C1 18 25 C7 D6 23 26 C3 97 24 :84
56C0 E6 21 CB 02 D7 25 86 01 8E 7F 26 A7 24 CC 3E 80 :DF
56D0 ED 84 86 04 A7 02 20 A9 C1 35 24 A5 D6 29 26 A1 :F2
56E0 97 2A E6 21 CB 02 D7 2B 86 02 8E 7F 2C 20 DC 81 :D5
56F0 76 24 8E C1 27 24 19 C1 18 25 86 D6 2F 10 26 FF :0B
-----------------------------------------------------------
Sum: 9D 6B 29 42 8E C2 A9 29 D7 F5 9B 45 3E 02 26 27 :CE
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5700 80 97 30 E6 21 CB 02 D7 31 86 03 8E 7F 32 20 BB :C6
5710 C1 35 10 24 FF 6B D6 35 10 26 FF 65 97 36 E6 21 :0D
5720 CB 02 D7 37 86 04 8E 7F 38 20 A0 8E 7F 23 86 06 :26
5730 5A 3D 3A 34 20 1F 12 6C A4 A6 21 C6 50 3D 1F 01 :A0
5740 E6 22 3A 30 89 40 00 A6 A4 81 18 26 0E 12 86 80 :6A
5750 97 46 EC 21 DD 52 35 30 16 FD D9 86 01 97 46 97 :65
5760 47 96 FD 26 04 8D 5C 35 A0 17 00 7D 96 FD 4A 26 :59
5770 10 A6 22 81 41 24 44 30 02 8B 02 A7 22 8D 44 35 :90
5780 A0 4A 26 12 A6 21 81 33 25 31 30 89 FD D0 80 07 :00
5790 A7 21 8D 2F 35 A0 4A 26 10 A6 22 81 0C 25 1C 30 :9F
57A0 1E 80 02 A7 22 8D 1C 35 A0 A6 21 81 84 24 0C 30 :13
57B0 89 02 30 8B 07 A7 21 8D 0A 35 A0 6F A4 35 20 6F :58
57C0 24 20 26 34 10 EE 23 6A 25 26 09 86 04 A7 25 33 :06
57D0 C8 34 EF 23 86 0D 97 1E EC C1 ED 84 EC C1 ED 02 :10
57E0 30 88 50 0A 1E 26 F1 35 90 34 10 CE 00 00 86 0D :B1
57F0 C6 50 EF 84 EF 02 3A 4A 26 F8 35 90 A6 A4 C6 50 :41
-----------------------------------------------------------
Sum: 0A C8 CF C5 18 B4 3A 54 1F 57 04 79 73 55 2B BD :63
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5800 3D 1F 01 E6 21 3A A6 22 6C 22 4A 26 13 86 02 97 :96
5810 48 97 46 86 06 97 55 17 01 42 17 01 1A 16 00 CF :0E
5820 4A 26 06 17 01 6B 16 00 C6 4A 26 3D 17 01 2D 17 :DE
5830 00 BD C6 50 A6 23 81 02 26 16 30 89 00 A0 CE FC :7E
5840 3F EF 84 3A CE F0 0F EF 84 3A CE FC 3F EF 84 39 :1B
5850 30 89 01 40 CE F0 0F EF 84 3A CE 00 00 EF 84 3A :EF
5860 EF 84 3A CE F0 0F EF 84 39 4A 26 0D 17 01 22 17 :F4
5870 00 C5 30 89 40 00 16 00 BE 6F A4 6F 26 96 F0 81 :41
5880 64 27 10 96 51 9B F0 81 0A 24 04 97 51 20 20 80 :68
5890 0A 97 51 96 50 4C 81 0A 24 04 97 50 20 11 0F 50 :4E
58A0 96 4F 4C 81 0A 24 04 97 4F 20 04 0F 4F 0C 4E 96 :3C
58B0 4E 8E BA E3 8D 1E 30 89 FD D1 96 4F 8D 16 30 89 :EC
58C0 FD D1 96 50 8D 0E 30 89 FD D1 96 51 8D 06 30 89 :09
58D0 FD D1 86 00 C6 08 8B 30 3D C3 D8 00 1F 03 86 07 :64
58E0 97 1E C6 50 A6 C0 43 A7 84 3A 0A 1E 26 F6 39 34 :8A
58F0 30 A6 23 31 89 40 00 81 03 27 19 CE D2 40 86 06 :23
-----------------------------------------------------------
Sum: 40 5B 6E 05 54 8D 58 29 93 FF E3 E7 B1 44 39 3D :37
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5900 97 1E EC C1 ED 84 ED A4 30 88 50 31 A8 50 0A 1E :BD
5910 26 F0 35 B0 CE D2 4C 30 1F 31 3F 86 0B 97 1E EC :D8
5920 C1 ED 84 ED A4 EC C1 ED 02 ED 22 30 88 50 31 A8 :4F
5930 50 0A 1E 26 EA 35 B0 34 10 A6 23 C6 50 CE 00 00 :5E
5940 81 03 27 0A 86 06 EF 84 3A 4A 26 FA 35 90 30 1F :6C
5950 86 0B EF 84 EF 02 3A 4A 26 F8 35 90 34 10 8E 5E :8C
5960 77 86 07 97 1E 0F 1F 0C 1F 86 01 8D 0C 8D 0A 08 :D1
5970 1F 0C 1F 0A 1E 26 F4 35 90 C6 02 D7 20 D6 1F E7 :EC
5980 84 E7 86 30 88 50 0A 20 26 F5 8B 02 30 89 00 9F :23
5990 39 34 10 8E 5E 77 86 07 97 1E 0F 1F 86 01 8D D9 :3D
59A0 8D D7 0A 1E 26 F8 35 90 0A FF 10 26 FC CC 96 FE :0A
59B0 97 FF 86 51 C6 20 DD 1E 96 FD 27 27 91 FC 27 04 :E7
59C0 97 FC 20 1F 4A 26 06 86 2D 97 1F 20 16 4A 26 06 :5D
59D0 86 36 97 1E 20 0D 4A 26 06 86 12 97 1F 20 04 86 :0C
59E0 6D 97 1E 96 21 84 1F 9B 1E A7 A4 96 21 49 84 0F :13
59F0 9B 1F A7 21 6F 22 6F 23 6F 26 86 0A A7 27 A7 28 :67
-----------------------------------------------------------
Sum: 71 7E A1 D4 C6 6C 66 43 8D D3 5E 60 60 34 DF 5B :2B
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5A00 16 FC 77 A6 A4 C6 50 3D 1F 01 E6 21 3A A6 26 10 :63
5A10 26 00 81 96 23 10 26 00 8F 96 29 10 26 00 89 96 :39
5A20 2F 10 26 00 83 96 35 26 7F A6 23 81 02 26 14 EC :CA
5A30 A4 81 5C 25 73 81 62 24 6F C1 26 25 6B C1 28 24 :13
5A40 67 20 16 81 03 26 61 EC A4 81 59 25 5B 81 60 24 :97
5A50 57 C1 25 25 53 C1 2A 24 4F 7F D3 60 6C 26 34 30 :BB
5A60 8E 16 04 10 8E 56 04 CE 3F B8 86 07 97 1E EC C1 :54
5A70 ED 84 ED A4 EC C1 ED 02 ED 22 EC C1 ED 04 ED 24 :5C
5A80 EC C1 ED 06 ED 26 30 88 50 31 A8 50 0A 1E 26 DE :10
5A90 35 30 20 14 96 FD 26 08 0A FB 26 0C 86 14 97 FB :BD
5AA0 6F 26 86 01 A7 28 8D 17 EC A4 C1 48 24 0E C1 02 :1D
5AB0 27 0A 81 11 25 06 81 9A 10 25 F6 C6 6F A4 39 34 :7A
5AC0 30 8E 16 04 10 8E 56 04 86 07 C6 50 CE 00 00 EF :30
5AD0 84 EF A4 EF 02 EF 22 EF 04 EF 24 EF 06 EF 26 3A :63
5AE0 31 A8 50 4A 26 E9 35 B0 00 00 00 00 00 00 00 00 :67
5AF0 03 C0 0F F0 0E 70 01 80 07 E0 3F FC FF FF 31 8C :9E
-----------------------------------------------------------
Sum: E7 0E D3 14 22 12 9B CB A2 A3 A4 C9 0E 28 66 B3 :77
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5B00 FC 3F 02 40 01 80 1F F8 1F F8 00 FF FF 00 0F FF :38
5B10 FF F0 3F FF FF FC FC F3 CF 3F FF FF FF FF 0F 03 :33
5B20 C0 F0 0F F8 1F F0 00 04 20 00 00 02 40 00 00 01 :2D
5B30 80 00 00 03 C0 00 00 FF FF 00 00 FF FF 00 00 1F :5E
5B40 FF FF F8 00 00 FF FF FF FF 00 07 FF FF FF FF E0 :D5
5B50 1F FF FF FF FF F8 3F FF FF FF FF FC FF 8F C7 E3 :82
5B60 F1 FF FF FF FF FF FF FF FF FF FF FF FF FF 03 F0 :D7
5B70 07 E0 0F C0 FF F0 0F FF 00 18 18 00 00 0C 30 00 :1F
5B80 00 06 60 00 00 03 C0 00 00 07 E0 00 03 FF FF C0 :D1
5B90 03 FF FF C0 03 FF FF C0 00 00 FF FF FF FF 00 00 :7E
5BA0 00 0F FF FF FF FF F0 00 00 FF FF FF FF FF FF 00 :F5
5BB0 03 FF FF FF FF FF FF C0 0F FF FF FF FF FF FF F0 :B6
5BC0 3F FF FF FF FF FF FF FC FF F0 FF 0F F0 FF 0F FF :2F
5BD0 FF F0 FF 0F F0 FF 0F FF FF FF FF FF FF FF FF FF :F2
5BE0 FF FF FF FF FF FF FF FF 00 FF 00 0F F0 00 FF 00 :F5
5BF0 00 3C 00 03 C0 00 3C 00 0F FF E0 07 FF F0 00 00 :1F
-----------------------------------------------------------
Sum: 94 39 AF C6 8B 4F 5E 64 26 3F D7 1A 18 82 21 83 :72
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5C00 30 0C 00 00 00 00 18 18 00 00 00 00 0C 30 00 00 :A8
5C10 00 00 06 60 00 00 00 00 03 C0 00 00 00 00 0F F0 :28
5C20 00 00 00 1F FF FF F8 00 00 1F FF FF F8 00 00 1F :49
5C30 FF FF F8 00 00 1F FF FF F8 00 00 00 1F FF FF FF :27
5C40 FF F8 00 00 00 03 FF FF FF FF FF FF C0 00 00 3F :F3
5C50 FF FF FF FF FF FF FC 00 00 FF FF FF FF FF FF FF :EF
5C60 FF 00 03 FF FF FF FF FF FF FF FF C0 0F FF FF FF :C6
5C70 FF FF FF FF FF F0 3F FF FF FF FF FF FF FF FF FC :1E
5C80 FF F8 1F F8 1F F8 1F F8 1F FF FF F8 1F F8 1F F8 :7F
5C90 1F F8 1F FF FF F8 1F F8 1F F8 1F F8 1F FF FF FF :8D
5CA0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
5CB0 FF FF 00 3F F0 00 1F F8 00 0F FC 00 00 1F E0 00 :4E
5CC0 0F F0 00 07 F8 00 00 07 80 00 03 C0 00 01 E0 00 :29
5CD0 0F 0F 03 C0 FF F3 0F FC 3F F0 0F 1C 00 F0 F0 00 :18
5CE0 00 FF 38 F0 0F F8 FF 00 00 0F F0 FF FF 1F FF F0 :38
5CF0 0F FF FF 00 0F FF FF FC 3F FF FF F0 00 FF FF 00 :41
-----------------------------------------------------------
Sum: 74 EC 76 68 1E E8 B1 FA 33 DE 15 76 2C 50 D6 2D :0A
```

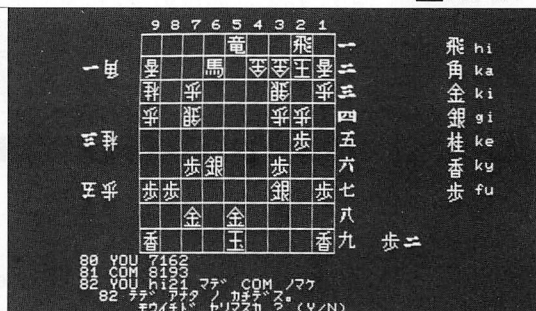
```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5D00 07 0F 00 18 00 0F F0 00 00 00 00 00 00 00 00 00 :2D
5D10 00 00 00 00 00 00 00 00 00 07 C0 00 00 1F F0 00 :D6
5D20 00 7F FC 00 00 1F F0 00 00 07 C0 00 00 00 00 00 :51
5D30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5D40 00 00 00 00 00 00 00 00 00 0F E0 00 00 7F FC 00 :6A
5D50 00 FF FE 00 01 FF FF 00 00 FF FE 00 00 7F FC 00 :74
5D60 00 0F E0 00 00 00 00 00 00 00 00 00 00 00 00 00 :EF
5D70 00 00 00 00 00 00 00 00 00 0F E0 00 00 FF FE 00 :EC
5D80 01 FF FF 00 03 FF FF 80 07 FF FF C0 03 FF FF 80 :C6
5D90 01 FF FF 00 00 FF FE 00 00 0F E0 00 00 00 00 00 :EB
5DA0 00 00 00 00 00 00 00 00 00 01 00 00 00 FF FE 00 :FE
5DB0 03 FF FF 80 07 FF FF C0 0F FF FF E0 1F FF FF F0 :40
5DC0 0F FF FF E0 07 FF FF C0 03 FF FF 80 00 FF FE 00 :30
5DD0 00 01 00 00 00 00 00 00 00 01 00 00 00 7F FC 00 :7D
5DE0 03 FF FF 80 0F FF FF E0 1F FF FF F0 3F FF FF F8 :B0
5DF0 7F FF FF FC 3F FF FF F8 1F FF FF F0 0F FF FF E0 :A8
-----------------------------------------------------------
Sum: 9D 97 D4 F4 60 27 D8 D8 57 37 19 00 70 95 DA 48 :01
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5E00 03 FF FF 80 00 7F FC 00 00 01 00 00 00 FF FE 00 :FA
5E10 07 FF FF C0 1F FF FF F0 7F FF FF FC 7F FF FF FC :C4
5E20 FF FF FF FE FF FF FF FF FF FF FF FE 7F FF FF FC :6B
5E30 7F FF FF FC 1F FF FF F0 07 FF FF C0 00 FF FE 00 :48
5E40 30 00 0F F0 03 F0 30 0C 30 00 30 0C 0C 0C 30 30 :42
5E50 30 00 30 0C 30 00 30 C0 30 00 30 0C 30 00 3F 00 :67
5E60 30 00 30 0C 30 00 30 C0 30 00 30 0C 0C 0C 30 30 :70
5E70 3F FC 0F F0 03 F0 30 0C 00 00 00 00 00 00 00 00 :69
5E80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5E90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5EA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5EB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5EC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5ED0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5EE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5EF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-----------------------------------------------------------
Sum: 57 F8 7A 32 A3 5C B9 77 15 FE 8D DE 46 14 99 58 :F3
```

FM-7/11

◉おまちかね./ 将棋対局FM版◉

将棋対局

■MAX

I/O'83年 9 月号のPC-6001将棋対局をKコンパイラに移植できましたので発表します.

遊び方

遊び方は 9 月号のPC-6001 将棋対局と同じです.

EXEC&H3500でスタートし先手をとるか後手をとるか聞いてきます. 先手なら[1], 後手なら[2]を入力し[↵]を押すと将棋盤が表示され, ゲームが始まります.

手の入力フォーマットは図 3 を見てください. 誤入力したときは[⇦]でバック・スペースが使えますし[C]を入力すれば最初から入力のやりなおしができます. 途中でゲームを止めるときは, [BASIC]を押すと, ゲームを再度始めるか止めるか, 聞いてくるので, [Y]か[N]で答えてください.

プログラムについて

プログラムはサブ・プログラムと, メイン・プログラムの 2 つに分かれていますが, 内容はPC-6001将棋対局と同じです. Kコンパイラはランタイム・ルーチンをコンパイラが呼びだしているので, ランタイム・ルーチンの場所を変えることができません. 長いプログラムをコンパイルすることができないので, 2 つに分けて, KのプログラムからKのプログラムを呼び出すようになっています. 詳しいことはソース・リストを見てください.

プログラムは&H 1200 から&H 59 FFまでで, 手の入力フォーマット, コントロール・コマンドもPC-6001 将棋対局と同じです. 英字は大文字で打っても小文字になります.

プログラムはEXAS BASICコンパイラのソースをKコンパイラ用に書き直していますが, ソースが 20 Kバイト近くになってしまったので, メインとサブに分けてコンパイルし, 変数エリアとスタック・エリアを別々においてリンクします.

コンパイルについて

コンパイルは裏RAMにコンパイラをリロケートしてコンパイルしていますがワークが大きいのと, ラベルが多いのでワーク・エリアも裏RAMを使うようにして, ラベル・テーブルも多く設定しないとDOUBLE DEF ERRORがでます. 今回は**表 1** のように変更しています.

コンパイルのやり方

以下の手順でコンパイルしてください.

①ファイル版のKコンパイラV3.6の＄2082番地を＄10にする.

②**リスト 1**, **リスト 2** をそれぞれASCIIセーブしておく(**SAVE**"ファイル名", **A**). ただし, 11の場合変更があるので, FM-11用変更箇所に従って変更しておく.

③それぞれをコンパイルし, オブジェクトをセーブする. **リスト 2** のオブジェクトは, **表 1** の箇所を変更してからセーブする.

④**LOADM "MAIN", −&H1B00**
LOADM "SUB", −&H3C00
としてふたたびよびこむ.

⑤ダンプ・リストの$1200～$13 CDと$5000～$59 FFまでを打ち込む.

⑥**SAVEM"SHOGI", &H1200, &H59FF, &H3500**
で, できあがりです.

Kコンパイラについて

サブ・プログラム中の 7300 行と 7420 行の**FOR−NEXT**のところを**REPEAT−UNTIL**にしてありますが, **FOR**

図1　メモリ・マップ

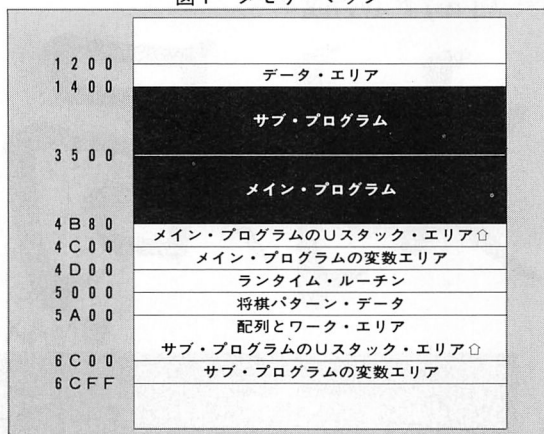



表1　サブ・プログラムのオブジェクトの変更内容

アドレス	旧	新
5003	4C	6C
5007	4C	6C

11変更箇所

```
15001 'HLT[];CPOKE ADDR,1,7,40,25,0,25,0
,1;SUB[]

15610 'HLT[];CPOKE ADDR,1,7,40,25,20,5,0
,1;SUB[]
```

G=8TO14とするとGの値が8～120のループになってしまいます．色々調べてみましたが原因がわからずREPEAT-UNTILだとだいじょうぶなようなのでそれですませています．たぶんUスタックのところが原因だとはおもうのですが．

なお，$3F16，7の$FFFFを0000に書き換えると，マイコンの動きを見ることができます．

図2　コントロール・コマンド

C	：入力のキャンセル．
BREAK	：ゲームの中止．
B	：2手戻す．
⇦	：バック・スペース．

図3　手の入力フォーマット

駒の移動

2726

2七にある駒　を　2六に移動する

移動後，成る

5453n（小文字のnを付ける）

5四の駒を5三に移動し，成る

駒を打つ

hi52

飛車　5二に打つ

以上は RETURN 入力後，実行される．

参考文献

1) "FM-8活用研究"，工学社
2) "FM-7/8活用研究"，工学社
3) 黒崎　隆："PC-6001『将棋対局』"，I/O，83年9月号
4) COMPAC S："FM-7拡張Kコンパイラ" I/O，83年3月号
5) コンピ："FM-7/8 Kコンパイラ用グラフィック・サブルーチン"，I/O，83年4月号

Kコンパイラでfor-nextを使うときの注意（編集部）

Kコンパイラのfor文では，終値（TOの後に書く値）がUスタックに積まれます．したがって，FOR-NEXTの中から外へ，GOTO文やRETURN文で飛び出すと，Uスタックの値がすべて，異常な動作をします．

ループからの飛び出しがあるときは，FOR-NEXT文でなく，WHILE-WEND文またはREPEAT-UNTIL文を使ってください．

リスト1　メイン・プログラム

```
0 'DX[$1E1,$6800];POKE $1E1,$4F;POKE $1E2,$30
10 'L10;GOSUB L60000
100 'SUBSYS=$1400
210 'B=$5A00;M=$5B20;GK=$5B40;GQ=$5B50;LB=$5B90;
LK=$5C00;BA=$5C70;BB=$5C90;BC=$5CB0;BD=$5CD0;BE=
$5CF0;BF=$5D10;PS=$5D30
230 'PW=$5D46;DN=$5DE8;Q=$5E00;QK=$5E20;R=$5E40;
RK=$5E60;RV=$5E80;S=$5EA0;TE=$5EE0;TF=$5F10;TG=$
5F40;TA=$5F70
250 'QD=$5FA0;RD=$5FC0;QM=$5FE0;RM=$6000;PK=$602
0;PN=$6060;UT=$6090;UU=$60B0;UV=$60D0;UW=$6100;U
Y=$6130;UZ=$6160
260 'PV=$6190;PX=$61C0;PY=$61F0;PZ=$6220;STR=$62
50;GS=$6300
300 'GOTO L15000
500 'L500;POKE $6260,6;CALL SUBSYS;RETURN
3000 'L3000;BN=21-X+11*Y;RETURN
3100 'L3100;Y=(BN-11)/11
3110 'X=(21-BN)+11*Y;RETURN
3200 'L3200;IF (X<-3)+(12<X)+(Y<1)+(9<Y) THEN NG
=-1;PRINT" 3210";RETURN;FI
3220 'L3220;AE=$5000+52*K
3225 'X9=(12-X)*33;Y9=Y*165/10-4
3230 'LINE[X9,Y9,X9+31,Y9+12,0,0,2]
3240 'HLT[];AD=$FC82;CPOKEAD,$1C,X9#,Y9#,(X9+31)
#,(Y9+12)#,7,0,52
3250 'FOR I=0 TO 51;AD:I)=AE:I);NEXT;SUB[];RETUR
N
3300 'L3300;
3310 'X=0;FOR Y=1 TO 9;K=Y+29;GOSUB L3200;NEXT
3320 'FOR I=0 TO 9
3330 'W=33*I;X=98+W;Y=10+W/2
3340 'LINE[98,Y,395,Y,0,7,0]
3350 'LINE[X,10,X,158,0,7,0]
3360 'NEXT ;LOC [0,0]
3370 'PRINT "        9 8 7 6 5 4 3 2 1",/
3375 'X=-5;Y=1;II=2;K=1;FOR I1=1 TO 7;LOC [38,II
];PRINT CHR$(GK:II)),CHR$(GK:II+1));GOSUB L3220;
Y=Y+1;II=II+2;K=K+1;NEXT
3380 'LOC [0,20];RETURN
3400 'L3400;IF DS THEN GOTO L3430;FI
3405 'GOSUB L3100;GOSUB L3200
3430 'L3430;B(BN)=K;RETURN
3500 'L3500;FOR I=0 TO 21;B(I)=25;NEXT
3510 'FOR J=22 TO 110 STEP 11
3520 'B(J)=25;FOR I=1 TO 9;B(I+J)=10;NEXT;B(J+10
)=25;NEXT
3540 'FOR I=121 TO 142;B(I)=25;NEXT;RETURN
3600 'L3600;GOSUB L3500;KDATA=$6500
3610 'FOR JJ=110 TO 88 STEP -11
3620 'FOR II=1 TO 9
3630 'K=KDATA:0);KDATA=KDATA+1;IF K=10 THEN GOTO
 L3670;FI
3640 'BN=II+JJ;GOSUB L3400
3650 'K=K+15;BN=142-BN;GOSUB L3400
3670 'L3670;NEXT;NEXT;RETURN
4000 'L4000;
4010 'L4010;I0=0;PRINT/;PRINT N," YOU ",
4020 'L4020;IK=GET;IF PEEK($4F34)<>0 THEN GOTO G
END;FI;IF IK=0 THEN GOTO L4020;FI
4025 'IF (64<IK)*(IK<91) THEN IK=IK+32;FI
4030 'IF IK=8 THEN IF I0>0 THEN I0=I0-1;PRINT CH
R$($1D)," ",CHR$($1D);FI;GOTO L4020;FI
4040 'IF IK=13 THEN RETURN;FI
```

リスト1　メイン・プログラム

```
4050 'IF IK=99 THEN GOTO L4010;FI
4060 'POKE $6250+I0,IK;I0=I0+1;PRINT CHR$(IK)
4070 'GOTO L4020
4600 'L4600;POKE $6260,2;CALL SUBSYS;RETURN
4700 'L4700;POKE $6260,4;CALL SUBSYS;RETURN
5500 'L5500;POKE $6260,21;CALL SUBSYS;RETURN
5600 'L5600;POKE $6260,3;CALL SUBSYS;RETURN
5700 'L5700;POKE $6260,1;CALL SUBSYS;RETURN
5900 'L5900;POKE $6260,18;CALL SUBSYS;RETURN
6200 'L6200;POKE $6260,22;CALL SUBSYS;RETURN
6300 'L6300;POKE $6260,17;CALL SUBSYS;RETURN
6700 'L6700;POKE $6260,5;CALL SUBSYS;RETURN
6800 'L6800;POKE $6260,7;CALL SUBSYS;RETURN
7000 'L7000;POKE $6260,8;CALL SUBSYS;RETURN
7200 'L7200;POKE $6260,9;CALL SUBSYS;RETURN
7600 'L7600;POKE $6260,11;CALL SUBSYS;RETURN
7700 'L7700;POKE $6260,12;CALL SUBSYS;RETURN
8700 'L8700;POKE $6260,10;CALL SUBSYS;RETURN
8800 'L8800;POKE $6260,14;CALL SUBSYS;RETURN
8900 'L8900;POKE $6260,16;CALL SUBSYS;RETURN
9000 'L9000;POKE $6260,15;CALL SUBSYS;RETURN
9300 'L9300;POKE $6260,19;CALL SUBSYS;RETURN
9500 'L9500;POKE $6260,20;CALL SUBSYS;RETURN
9600 'L9600;POKE $6260,13;CALL SUBSYS;RETURN
15000 'L15000;
15001 'HLT[];CPOKE ADDR,1,0,40,25,0,25,0,1,0;SUB
[]
15002 'PRINT " ｼｳｷﾞ ﾀｲｷｮｸ  Ver 2.0 by MAX",/,/
15003 'PRINT " ｾﾝﾃ ... 1",/,/," ｺﾞﾃ ... 2 ";CH=I
NPUT
15010 'HLT[];CPOKE ADDR,1,0,40,25,20,5,0,1,0;SUB
[]
15020 'GOSUB L3300;GOSUB L3600
15030 'N=0;DX[ADR(N),$6CA8];FOR I=1 TO 30;GQ(I)=
0;NEXT
15050 'IF CH=2 THEN GOTO L17000;FI
16000 'L16000;PRINT CHR$(7);C=0;DX[ADR(C),$6C98]
;N=N+1;DX[ADR(N),$6CA8]
16020 'L16020;GOSUB L4000;DX[ADR(I0),$6C9E];NG=0
;DX[ADR(NG),$6C7C]
16030 'IF STR:0)=98 THEN IF N<3 THEN PRINT CHR$(
7)," ｺﾚｲｼﾞｮｳﾓﾄﾞﾚﾏｾﾝ";ELSE;GOSUB L5700;GOSUB L570
0;DX[$6CA8,ADR(N)];N=N-2;DX[ADR(N),$6CA8];FI;GOT
O L16020;FI
16040 'GOSUB L4600;DX[$6C7C,ADR(NG)];IF NG=-1 TH
EN PRINT CHR$(7);GOSUB DSP;GOTO L16020;FI
16050 'GOSUB L5600;DX[$6C7C,ADR(NG)];IF NG=-1 TH
EN PRINT CHR$(7);GOSUB DSP;GOTO L16020;FI
16060 'DX[$6C6A,ADR(F)];IF B(F)=15 THEN PRINT "
YOU ﾉ ｶﾁ";GOTO GEND;FI
16070 'GOSUB L4700;GOTO L17000
17010 'L17000;C=1;DX[ADR(C),$6C98];N=N+1;DX[ADR(
N),$6CA8];TP=0;DX[ADR(TP),$6CBA];ZN=0;DX[ADR(ZN)
,$6CF0];DS=-1;DX[ADR(DS),$6C88]
17020 'FOR I=1 TO 20;PV(I)=0;NEXT
17100 'K=0;DX[ADR(K),$6C56];GOSUB L6700;DX[$6C5A
,ADR(BN)];O0=BN;DX[ADR(O0),$6CF2];DX[$6C7C,ADR(N
G)];IF NG=-1 THEN GOTO GEND;FI
17110 'DX[$6CF2,ADR(O0)];F=O0;DX[ADR(F),$6C6A];G
OSUB L500;DX[$6C60,ADR(RN)];IF RN=0 THEN GOTO L1
7200;FI
17120 'PRINT " COM ﾉ ｶﾁ"
17140 'PRINT/,"  ",N-1," ﾃﾃﾞ COM ﾉ ｶﾁﾃﾞｽ｡";GOTO
GEND
17200 'L17200;K=15;DX[ADR(K),$6C56];GOSUB L6700;
DX[$6C5A,ADR(BN)];O1=BN;DX[ADR(O1),$6CAE];DX[$6C
7C,ADR(NG)];IF NG=-1 THEN GOTO GEND;FI
17210 'DX[$6CAE,ADR(O1)];F=O1;DX[ADR(F),$6C6A]:G
OSUB L500
17220 'DX[$6C58,ADR(QN)];IF QN=0 THEN GOTO L1740
0;FI
17230 'GOSUB L6800;GOSUB L7000;GOSUB L7200
17240 'DX[$6CBA,ADR(TP)];IF TP=0 THEN PRINT " ﾏﾃ
ﾞ COM ﾉﾏｹ ";GOTO L17260;FI
17250 'GOTO L19400
17260 'L17260;
17270 'PRINT/,"  ",N-1," ﾃﾃﾞ ｱﾅﾀ ﾉ ｶﾁﾃﾞｽ｡";GOTO
GEND
17400 'L17400;GOSUB L8700;UA=0;DX[ADR(UA),$6CF4]
;DX[$6CC6,ADR(UX)];IF UX=0 THEN GOTO L17500;FI
17405 'DX[$6CC6,ADR(UX)];IF UX=1 THEN GOTO L1742
0;FI
17410 'GOSUB L7600
17420 'L17420;DX[$6CF4,ADR(UA)];UA=UA+1;DX[ADR(U
A),$6CF4];DX[$6CC6,ADR(UX)];IF (UA>UX)+(UZ(UA)<0
) THEN GOTO L17500;FI
17430 'HI=0;DX[ADR(HI),$6C9C];E=UW(UA);DX[ADR(E)
,$6C6C];F=UY(UA);DX[ADR(F),$6C6A]
17440 'GOSUB L7700;DX[$6CAC,ADR(OT)];IF OT=-1 TH
EN GOTO L17420;FI
17450 'GOSUB L9600;DX[$6C7C,ADR(NG)];IF NG=-1 TH
EN NG=0;DX[ADR(NG),$6C7C];GOTO L19400;FI
17460 'GOTO L17420
17500 'L17500;GOSUB L8800;DX[$6CC6,ADR(UX)];IF U
X=0 THEN GOTO L18000;FI
17510 'DX[$6CC6,ADR(UX)];IF UX=1 THEN GOTO L1753
0;FI
17520 'GOSUB L7600
17530 'L17530;DX[$6CC6,ADR(UX)];IF UZ(UX)>=0 THE
N GOTO L18000;FI
17535 'GOSUB L9000
17540 'DX[$6CDC,ADR(FR)];IF FR=-1 THEN FR=0;DX[A
DR(FR),$6CDC];GOTO L19400;FI
17600 'GOSUB L8900
17610 'DX[$6CDC,ADR(FR)];IF FR=-1 THEN FR=0;DX[A
DR(FR),$6CDC];GOTO L18000;FI
17620 'GOSUB L9600;DX[$6C7C,ADR(NG)];IF NG=-1 TH
EN NG=0;DX[ADR(NG),$6C7C];GOTO L19400;FI
18000 'L18000;GP=0;DX[ADR(GP),$6CAA]
18010 'L18010;DX[$6CAA,ADR(GP)];GP=GP+1;DX[ADR(G
P),$6CAA];DX[$6CF6,ADR(IN)];IF GP>18 THEN GOTO L
18500;FI
18015 'DX[$6CAA,ADR(GP)];IF GQ(GP)=-1 THEN GOTO
L18010;FI
18020 'GOSUB L6300;DX[$6C7C,ADR(NG)];IF NG=-1 TH
EN NG=0;DX[ADR(NG),$6C7C];GOTO L18010;FI
18030 'GOSUB L9600;DX[$6C7C,ADR(NG)];IF NG=-1 TH
EN NG=0;DX[ADR(NG),$6C7C];GOTO L19400;FI
18040 'DX[$6CBA,ADR(TP)];PV(TP)=1
18500 'L18500;GOSUB L5900
18510 'GOSUB L9600;DX[$6CBA,ADR(TP)];IF TP>2 THE
N NG=0;DX[ADR(NG),$6C7C];GOTO L19400;FI
18520 'GOTO L18500
19400 'L19400;GOSUB L9300
19410 'GOSUB L9500;DX[$6CDC,ADR(FR)];IF FR=0 THE
N GOTO L19420;FI
19413 'DX[$6CF0,ADR(ZN)];IF ZN>3 THEN PRINT " ﾏﾃ
ﾞ COM ﾉ ﾄｳﾘｮｳ";GOTO L17260;FI
19415 'DX[$6CF0,ADR(ZN)];ZN=ZN+1;DX[ADR(ZN),$6CF
0];TP=0;DX[ADR(TP),$6CBA];GOTO L18500
19420 'L19420;HI=-1;DX[ADR(HI),$6C9C];DX[$6CEA,A
DR(IW)];IF TG(IW)=-1 THEN HI=0;DX[ADR(HI),$6C9C]
;FI
19430 'DX[$6CEA,ADR(IW)];E=TE(IW);DX[ADR(E),$6C6
C];F=TF(IW);DX[ADR(F),$6C6A];NA=TA(IW);DX[ADR(NA
),$6C92];G=TG(IW);DX[ADR(G),$6C94]
19440 'DX[$6CEA,ADR(IW)];IF PV(IW)=1 THEN DX[$6C
AA,ADR(GP)];GQ(GP)=-1;FI
19500 'GOSUB L5500;DX[$6C7C,ADR(NG)];IF NG=-1 TH
EN NG=0;DX[ADR(NG),$6C7C];NA=0;DX[ADR(NA),$6C92]
;GOTO L19520;FI
19510 'NA=-1;DX[ADR(NA),$6C92]
19520 'L19520;GOSUB L6200
19525 'DS=0;DX[ADR(DS),$6C88]
19530 'GOSUB L4700
19540 'GOTO L16000
30000 'DX;ADDR=%2;DATA=PEEK(%1)*256+PEEK(%1+1);C
POKE ADDR,DATA#;RETURN
30360 'CLS;HLT[];CPOKE ADDR,$02,%1,%3,%2;SUB[];R
ETURN
30370 'LINE;HLT[];CPOKE ADDR,$15,%6,%5,%1#,%2#,%
3#,%4#,%7;SUB[];RETURN
30400 'HLT;CODE $B6,$FD05#,$2BFB#;POKE $FD05,$80
;CODE $B6,$FD05#,$2AFB#;ADDR=$FC82;RETURN
30410 'SUB;POKE $FD05,0;RETURN
30500 'LOC;HLT[];CPOKE ADDR,3,3,$12,%1,%2;SUB[];
RETURN
40000 'GEND;PRINT /,"       ﾓｳｲﾁﾄﾞ ﾔﾘﾏｽｶ ? (Y/N)"
40010 'START;A=GET;IF (A=89)+(A=121) THEN GOTO L
10;ELSE;IF (A=78)+(A=110) THEN DX[$6800,$1E1];EN
D;FI;FI;GOTO START
50000 'DSP;DA=PEEK($6270)
50010 'IF DA=1 THEN PRINT" ｺﾏｷｺﾞｳﾁｶﾞｲ";FI
50020 'IF DA=2 THEN PRINT" ﾌｫｰﾏｯﾄ･ｴﾗｰ";FI
50030 'IF DA=3 THEN PRINT" ｼﾞﾌﾞﾝﾉｺﾏﾃﾞﾅｲ";FI
50040 'IF DA=4 THEN PRINT" ｼﾞﾌﾞﾝﾉｺﾏｦﾄｯﾃｲﾙ";FI
50050 'IF DA=5 THEN PRINT" ｷｲﾃﾅｲ";FI
50060 'IF DA=6 THEN PRINT" ﾆﾌ";FI
50070 'IF DA=7 THEN PRINT" ｳﾂｺﾏｶﾞﾅｲ";FI
50080 'IF DA=8 THEN PRINT" ｳﾂﾊﾞｼｮﾆｺﾏｱﾘ";FI
50090 'IF DA=9 THEN PRINT" ｼｺﾞﾏ";FI
50100 'IF DA=10 THEN PRINT" ﾅﾚﾅｲ";FI
50200 'RETURN
60000 'L60000;CODE $8E,$4C,0,$CC,0,0,$ED,$81,$8C
,$4D,0,$26,$F9,$8E,$5A,0,$ED,$81,$8C,$66,0,$26,$
F9,$8E,$6C,0,$ED,$81,$8C,$6D,0,$26,$F9
61010 'FOR I=0 TO 53;POKE $6500+I,PEEK($1200+I);
NEXT
```

リスト１　メイン・プログラム

```
61020 'FOR I=0 TO 15;POKE $5B40+I,PEEK($1240+I);
NEXT
61030 'FOR I=0 TO 9
61040 'FOR J=0 TO 15
61050 'A=I*16+J;POKE $5D46+A,PEEK($1250+A)
61060 'NEXT;NEXT
61070 'FOR I=0 TO 19;POKE $5DE8+I,PEEK($12F0+I);
NEXT
61090 'FOR I=0 TO 71
61110 'POKE $6304+I,PEEK($1310+I)
61160 'NEXT
61180 'FOR I=0 TO 59;POKE $6020+I,PEEK($1360+I);
NEXT
61190 'FOR I=0 TO 45;POKE $6060+I,PEEK($13A0+I);
NEXT
62000 'POKE $4F34,0;RETURN
```

リスト２　サブ・プログラム

```
210 'B=$5A00;M=$5B20;GK=$5B40;GQ=$5B50;LB=$5B90;
LK=$5C00;BA=$5C70;BB=$5C90;BC=$5CB0;BD=$5CD0;BE=
$5CF0;BF=$5D10;PS=$5D30
230 'PW=$5D46;DN=$5DE8;Q=$5E00;QK=$5E20;R=$5E40;
RK=$5E60;RV=$5E80;S=$5EA0;TE=$5EE0;TF=$5F10;TG=$
5F40;TA=$5F70
250 'QD=$5FA0;RD=$5FC0;QM=$5FE0;RM=$6000;PK=$602
0;PN=$6060;UT=$6090;UU=$60B0;UV=$60D0;UW=$6100;U
Y=$6130;UZ=$6160
260 'PV=$6190;PX=$61C0;PY=$61F0;PZ=$6220;STR=$62
50;GS=$6300;GOTO  L15000
300 'L300;IF K>14 THEN GOTO L320;FI
310 'QN=QN+1;Q(QN)=BN;QK(QN)=K;QD(QN)=DI;QM(QN)=
VN;RETURN
320 'L320;RN=RN+1;R(RN)=BN;RK(RN)=K;RD(RN)=DI;RM
(RN)=VN;RETURN
400 'L400;IF K>14 THEN K1=K-15;DJ=DI;GOTO L420;F
I
410 'K1=K;DJ=7-DI
420 'L420;IF K1>10 THEN K1=3;FI
430 'PO=PW(K1*8+DJ);RETURN
500 'L500;QN=0;RN=0;DI=0
510 'L510;VN=1;V=DN(DI);BN=F+V;K=B(BN)
520 'IF K=25 THEN GOTO L610;FI
530 'IF K=10 THEN GOTO L560;FI
540 'GOSUB L400;IF PO=0 THEN GOTO L610;FI
550 'GOSUB L300;GOTO L610
560 'L560;VN=VN+1;BN=F+VN*V;K=B(BN)
570 'IF K=25 THEN GOTO L610;FI
580 'IF K=10 THEN GOTO L560;FI
590 'GOSUB L400;IF PO<>2 THEN GOTO L610;FI
600 'GOSUB L300
610 'L610;DI=DI+1;IF DI>7 THEN GOTO L630;FI
620 'GOTO L510
630 'L630;BN=F-23;K=B(BN);VN=1
640 'IF K=20 THEN GOSUB L300;FI
650 'BN=F-21;K=B(BN)
660 'IF K=20 THEN GOSUB L300;FI
670 'BN=F+21;K=B(BN)
680 'IF K=5 THEN GOSUB L300;FI
690 'BN=F+23;K=B(BN)
700 'IF K=5 THEN GOSUB L300;FI
710 'RETURN
800 'L800;IF K>14 THEN K1=K-15;DJ=7-DI;GOTO L820
;FI
810 'K1=K;DJ=DI
820 'L820;IF K1>10 THEN K1=3;FI
830 'PO=PW(K1*8+DJ);RETURN
900 'L900;K=B(E);IF (K=10)+(K=25) THEN SN=-1;RET
URN;FI
910 'SN=0;DI=0
920 'IF K=5 THEN GOTO L1090;FI
930 'IF K=20 THEN GOTO L1110;FI
940 'L940;VN=1;V=DN(DI);F1=E+V;KF=B(F1)
950 'IF KF=25 THEN GOTO L1050;FI
960 'GOSUB L800
970 'IF PO=0 THEN GOTO L1050;FI
980 'SN=SN+1;S(SN)=F1
990 'IF (PO=1)+(KF<>10) THEN GOTO L1050;FI
1000 'L1000;VN=VN+1;F1=E+VN*V;KF=B(F1)
1010 'IF KF=25 THEN GOTO L1050;FI
1020 'SN=SN+1;S(SN)=F1
1030 'IF KF<>10 THEN GOTO L1050;FI
1040 'GOTO L1000
1050 'L1050;DI=DI+1
1060 'IF DI>7 THEN RETURN;FI
1070 'GOTO L940
1080 'L1080;IF B(W)<>25 THEN SN=SN+1;S(SN)=W;FI;
RETURN
1090 'L1090;W=E-23;GOSUB L1080
1100 'W=E-21;GOSUB L1080;RETURN
1110 'L1110;W=E+23;GOSUB L1080
1120 'W=E+21;GOSUB L1080;RETURN
3000 'L3000;BN=21-X+11*Y;RETURN
3100 'L3100;Y=(BN-11)/11;INT=0
3110 'X=(21-BN)+11*Y;RETURN
3200 'L3200;IF (X<-3)+(12<X)+(Y<1)+(9<Y) THEN NG
=-1;RETURN;FI
3220 'AE=$5000+52*K
3225 'X9=(12-X)*33;Y9=Y*165/10-4
3230 'LINE[X9,Y9,X9+31,Y9+12,0,0,2]
3240 'HLT[];AD=$FC82;CPOKEAD,$1C,X9#,Y9#,(X9+31)
#,(Y9+12)#,7,0,52
3250 'FOR I=0 TO 51;AD:I)=AE:I);NEXT;SUB[];RETUR
N
3400 'L3400;IF DS THEN GOTO L3430;FI
3405 'GOSUB L3100;GOSUB L3200
3430 'L3430;B(BN)=K;RETURN
3800 'L3800;M(MN)=M(MN)+MM;IF DS THEN RETURN;FI
3810 'IF MN>7 THEN GOTO L3900;FI
3820 'X=-2;Y=MN+2
3830 'IF M(MN)=0 THEN K=10;GOTO L3850;FI
3840 'K=MN
3850 'L3850;GOSUB L3200
3860 'X=-3;IF M(MN)=0 THEN K=10;GOTO L3880;FI
3870 'K=M(MN)+29
3880 'L3880;GOSUB L3200;RETURN
3900 'L3900;X=11;Y=MN-7
3910 'IF M(MN)=0 THEN K=10;GOTO L3930;FI
3920 'K=MN+8
3930 'L3930;GOSUB L3200
3940 'X=12;IF M(MN)=0 THEN K=10;GOTO L3960;FI
3950 'K=M(MN)+38
3960 'L3960;GOSUB L3200;RETURN
4100 'L4100;W=B(F);MM=1
4110 'IF W<8 THEN MN=W+7;GOTO L4170;FI
4120 'IF W<15 THEN MN=W;GOTO L4170;FI
4130 'IF W<23 THEN MN=W-15;GOTO L4170;FI
4140 'IF W<30 THEN MN=W-22;GOTO L4170;FI
4150 'PRINT"ERRER";RETURN
4170 'L4170;P=P+PK(W);BA(I2)=MN;BB(I2)=-1;GOSUB
L3800;RETURN
4200 'L4200;BC(I2)=F;BD(I2)=B(F)
4210 'BE(I2)=E;BF(I2)=B(E)
4220 'I2=I2+1;IF I2=10 THEN I2=0;FI
4230 'BN=F;W=B(E)
4240 'IF NA=-1 THEN K=W+7;P=P-PN(W);GOTO L4260;F
I
4250 'K=W
4260 'L4260;GOSUB L3400
4270 'BN=E;K=10;GOSUB L3400
4280 'NA=0;RETURN
4300 'L4300;IF 14<G THEN PRINT"g>41";RETURN;FI
4310 'PS(I2)=P
4320 'BA(I2)=G;BB(I2)=1
4330 'BC(I2)=F;BD(I2)=10
4340 'BE(I2)=0;BF(I2)=25
4350 'I2=I2+1;IF I2=10 THEN I2=0;FI
4360 'IF 7<G THEN K=8+G;GOTO L4380;FI
4370 'K=G
4380 'L4380;BN=F;GOSUB L3400
4390 'MN=G;MM=-1;GOSUB L3800;RETURN
4400 'L4400;I=1;REPEAT
4410 'IF GK(I)=I1 THEN G=I+7*C;RETURN;FI
4420 'I=I+1;UNTIL I>7
4430 'NG=-1;POKE $6270,1;RETURN
4500 'L4500;X=HIGH(I22)-48
4510 'Y=LOW(I22)-48
4520 'IF (X<1)+(9<X)+(Y<1)+(9<Y) THEN NG=-1;RETU
RN;FI
4530 'GOSUB L3000;RETURN
4600 'L4600;I22=STR(1);GOSUB L4500;F=BN;IF NG=-1
 THEN GOTO L4670;FI
4610 'I22=STR(0);GOSUB L4500;E=BN;IF NG=-1 THEN
GOTO L4650;FI
4620 'HI=0;NA=0
4630 'IF I0>4 THEN IF STR:4)=110 THEN NA=-1;FI;F
I
4640 'RETURN
4650 'L4650;NG=0;HI=-1;I1=I22;GOSUB L4400;IF NG=
-1 THEN GOTO L4670;FI
4660 'RETURN
4670 'L4670;POKE $6270,2;RETURN
4700 'L4700;IF HI=-1 THEN GOSUB L4300;RETURN;FI
```

```
4710 'GOSUB L4800;RETURN
4800 'L4800;PS(I2)=P
4810 'IF B(F)=10 THEN BB(I2)=0;GOTO L4830;FI
4820 'GOSUB L4100
4830 'L4830;GOSUB L4200;RETURN
5000 'L5000;KI=0;GOSUB L900
5010 'FOR I=1 TO SN
5020 'IF S(I)=F THEN KI=-1;RETURN;FI
5030 'NEXT ;RETURN
5100 'L5100;IF (B(E)<15*C)+(B(E)>15*C+14) THEN P
OKE $6270,3;NG=-1;RETURN;FI
5110 'IF (B(F)<15*C)+(B(F)>15*C+14)+(B(F)=10)THE
N GOTO L5130;FI
5120 'POKE $6270,4;NG=-1;RETURN
5130 'L5130;GOSUB L5000
5140 'IF KI=0 THEN POKE $6270,5;NG=-1;RETURN;FI
5150 'NG=0;RETURN
5200 'L5200;W=F-11;X=W-11*(W/11)
5210 'W1=7;IF G=14 THEN W1=22;FI
5220 'I=X+22;REPEAT
5230 'IF B(I)=W1 THEN NG=-1;POKE $6270,6;RETURN;
FI
5240 'I=I+11;UNTIL I>120
5250 'NG=0;RETURN
5300 'L5300;IF (G=5)*(F<43) THEN NG=-1;RETURN;FI
5310 'IF ((G=6)+(G=7))*(F<32) THEN NG=-1;RETURN;
FI
5320 'IF (G=12)*(F>99) THEN NG=-1;RETURN;FI
5330 'IF ((G=13)+(G=14))*(F>110) THEN NG=-1;RETU
RN;FI
5340 'NG=0;RETURN
5400 'L5400;IF M(G)=0 THEN POKE $6270,7;NG=-1;RE
TURN;FI
5410 'IF B(F)<>10 THEN POKE $6270,8;NG=-1;RETURN
;FI
5420 'IF (G=14)+(G=7) THEN GOSUB L5200;IF NG=-1
THEN RETURN;FI;FI
5430 'IF (G>4+C*7)*(G<8+C*7) THEN GOSUB L5300;IF
 NG=-1 THEN GOTO L5450;FI;FI
5440 'NG=0;RETURN
5450 'L5450;POKE $6270,9;RETURN
5500 'L5500;W=B(E)
5510 'IF (0<W)*(W<8)*(W<>3)*((E<54)+(F<54)) THEN
 GOTO L5540;FI
5520 'IF (15<W)*(W<23)*(W<>18)*((E>88)+(F>88)) T
HEN GOTO L5540;FI
5530 'NG=-1;POKE $6270,10;RETURN
5540 'L5540;NG=0;RETURN
5600 'L5600;IF HI=-1 THEN GOSUB L5400;RETURN;FI
5610 'GOSUB L5100;IF NG=-1 THEN RETURN;FI
5620 'IF NA=-1 THEN GOSUB L5500;FI
5630 'RETURN
5700 'L5700;I2=I2-1;IF I2=-1 THEN I2=9;FI
5730 'P=PS(I2);MN=BA(I2);MM=BB(I2);GOSUB L3800
5740 'BN=BC(I2);K=BD(I2);IF BN=0 THEN GOTO L5760
;FI
5750 'GOSUB L3400
5760 'L5760;BN=BE(I2);K=BF(I2);IF BN=0 THEN GOTO
 L5780;FI
5770 'GOSUB L3400
5780 'L5780;RETURN
5800 'L5800;LN=0;BN=23
5810 'L5810;W=B(BN);IF (W>14)*(W<30)*(W<>25) THE
N LN=LN+1;LB(LN)=BN;LK(LN)=W;FI
5820 'BN=BN+1;IF BN<120 THEN GOTO L5810;FI
5830 'RETURN
5900 'L5900;GOSUB L5800
5910 'L5910;E=LB(RND(LN)+1)
5920 'GOSUB L900;IF SN<1 THEN GOTO L5910;FI
5930 'F=S(RND(SN)+1)
5940 'IF (14<B(F))*(B(F)<30) THEN GOTO L5910;FI
5950 'GOSUB L5500;IF NG=0 THEN NA=-1;GOTO L5970;
FI
5960 'NG=0;NA=0
5970 'L5970;RETURN
6000 'L6000;IF G>7 THEN GOTO L6020;FI
6010 'I1=GK(G);RETURN
6020 'L6020;I1=GK(G-7);RETURN
6100 'L6100;GOSUB L3100
6110 'I22=(X+48)*256+Y+48;RETURN
6200 'L6200;BN=F;GOSUB L6100;I3=I22
6210 'IF HI=-1 THEN GOSUB L6000;STR(0)=I1;STR(1)
=I3;I0=4;GOTO L6230;FI
6220 'BN=E;GOSUB L6100;STR(0)=I22;STR(1)=I3;I0=4
6230 'L6230;IF NA=-1 THEN STR:4)=110;I0=5;FI
6240 'PRINT/;PRINT N," COM ",
6245 'FOR I=0 TO I0-1;PRINT CHR$(STR:I)),;NEXT;R
ETURN
6300 'L6300;STR(0)=GS(GP*2);STR(1)=GS(GP*2+1);GO
SUB L4600;GOSUB L5600;IF NG=-1 THEN RETURN;FI
6320 'GOSUB L7700;IF OT=-1 THEN NG=-1;FI
6330 'RETURN
6700 'L6700;BN=23
6710 'L6710;IF B(BN)=K THEN NG=0;RETURN;FI
6720 'BN=BN+1
6730 'IF BN<120 THEN GOTO L6710;FI
6740 'NG=-1;RETURN
6810 'L6800;K=15;GOSUB L6700;O1=BN
6820 'F=O1;GOSUB L500
6830 'QS=QN;IF QN>1 THEN RETURN;FI
6840 'FW=Q(1);F=FW;GOSUB L500;IF RN=0 THEN RETUR
N;FI
6850 'RW=RN;FOR I=1 TO RN;RV(I)=R(I);NEXT
6860 'FOR II=1 TO RW
6870 'F=FW;E=RV(II);HI=0;NA=0
6890 'GOSUB L4800;F1=F;E1=E
6900 'K=15;GOSUB L6700;F=BN;GOSUB L500
6910 'IF QN>0 THEN GOTO L6930;FI
6920 'TP=TP+1;TG(TP)=-1;TE(TP)=E1;TF(TP)=F1;TA(T
P)=0
6930 'L6930;GOSUB L5700;NEXT;RETURN
7000 'L7000;E=O1;GOSUB L900;II=1
7010 'L7010;BW=S(II);WK=B(BW)
7020 'IF (14<WK)*(WK<30) THEN GOTO L7110;FI;F=BW
;GOSUB L500
7050 'IF QN>0 THEN GOTO L7110;FI;E=O1;NA=0;GOSUB
 L4800
7080 'F1=F;E1=E;GOSUB L500;IF QN>0 THEN GOTO L71
00;FI
7090 'TP=TP+1
7095 'TG(TP)=-1;TE(TP)=E1;TF(TP)=F1;TA(TP)=0
7100 'L7100;GOSUB L5700
7110 'L7110;IF II<SN THEN II=II+1;GOTO L7010;FI
7120 'RETURN
7210 'L7200;F=O1;GOSUB L500;VM=QM(1);DD=QD(1)
7220 'IF (QN>1)+(VM=1) THEN RETURN;FI
7250 'II=1;DV=DN(DD);F=O1+DV;GOSUB L500;IF RN<2
THEN GOTO L7300;FI
7270 'FOR I=1 TO RN;IF RK(I)=15 THEN GOTO L7290;
FI
7280 'TP=TP+1;TG(TP)=-1;TE(TP)=R(I);TF(TP)=F;TA(
TP)=0
7290 'L7290;NEXT
7300 'L7300;G=8;REPEAT
7310 'GOSUB L5400;IF NG=-1 THEN GOTO L7330;FI
7320 'TP=TP+1
7325 'TG(TP)=G;TF(TP)=F
7330 'L7330;G=G+1;UNTIL G>14
7350 'L7340;II=II+1;IF II>=VM THEN RETURN;FI
7370 'F=O1+DV*II;GOSUB L500;IF RN=0 THEN GOTO L7
340;FI
7380 'IF RN=1 THEN GOSUB L7500;GOTO L7420;FI
7400 'FOR I=1 TO RN;TP=TP+1
7410 'TG(TP)=-1;TE(TP)=R(I);TF(TP)=F;TA(TP)=0;NE
XT
7420 'L7420;G=8;REPEAT
7430 'GOSUB L5400;IF NG=-1 THEN GOTO L7450;FI;TP
=TP+1
7440 'TG(TP)=G;TF(TP)=F
7450 'L7450;G=G+1;UNTIL G>14;GOTO L7340
7500 'L7500;E=R(1);E1=E;F1=F
7510 'NA=0;GOSUB L4800;GOSUB L500
7530 'IF RN=0 THEN GOTO L7550;FI
7540 'TP=TP+1;TG(TP)=-1;TE(TP)=E1;TF(TP)=F1;TA(T
P)=0
7550 'L7550;GOSUB L5700;RETURN
7600 'L7600;FOR I=2 TO UX
7610 'W=UZ(I);UZ(0)=W;W1=UY(I);W2=UW(I);J=I-1
7620 'L7620;IF W<=UZ(J) THEN GOTO L7650;FI
7630 'UZ(J+1)=UZ(J);UY(J+1)=UY(J);UW(J+1)=UW(J);
J=J-1;GOTO L7620
7650 'L7650;UZ(J+1)=W;UY(J+1)=W1;UW(J+1)=W2;NEXT
;RETURN
7720 'L7700;GOSUB L4700;F1=F;K=15;GOSUB L6700
7740 'F=BN;GOSUB L500;OT=-1;IF QN=0 THEN OT=0;FI
7750 'GOSUB L5700;F=F1;RETURN
7810 'L7800;FOR I=2 TO WN;W1=UT(I);W2=UU(I);UT(0
)=W1;J=I-1
7820 'L7820;IF PK(W1)<=PK(UT(J)) THEN GOTO L7850
;FI
7840 'UT(J+1)=UT(J);UU(J+1)=UU(J);J=J-1;GOTO L78
20
7860 'L7850;UT(J+1)=W1;UU(J+1)=W2;NEXT;RETURN
7920 'L7900;PB=P;US=0;IF C1=0 THEN GOTO L8200;FI
7950 'L7950;F=FU;GOSUB L500;IF RN=0 THEN RETURN;
FI
7990 'WN=RN;FOR I=1 TO RN;UU(I)=R(I);UT(I)=RK(I)
;NEXT
```

リスト2　サブ・プログラム

```
8020 'IF RN=1 THEN GOTO L8030;FI;GOSUB L7800
8030 'L8030;UI=1
8060 'L8040;IF WN<UI THEN RETURN;FI;E=UU(UI);GOS
UB L5500
8070 'IF NG=-1 THEN NG=0;NA=0;GOTO L8090;FI;NA=-
1
8090 'L8090;HI=0;GOSUB L4700
8110 'K=15;GOSUB L6700;F=BN;GOSUB L500;F=FU
8120 'IF QN>0 THEN UI=UI+1;GOSUB L5700;GOTO L804
0;FI
8140 'US=US+1;UV(US)=P-PB;IF US=1 THEN UW(UX)=E;
FI
8200 'L8200;F=FU;GOSUB L500;IF QN=0 THEN RETURN;
FI
8240 'WN=QN;FOR I=1 TO QN;UU(I)=Q(I);UT(I)=QK(I)
;NEXT
8260 'IF QN=1 THEN GOTO L8280;FI;GOSUB L7800
8280 'L8280;UI=WN
8290 'L8290;IF UI=0 THEN RETURN;FI
8320 'E=UU(UI);GOSUB L5500;IF NG=-1 THEN NG=0;NA
=0;GOTO L8340;FI;NA=-1
8340 'L8340;HI=0;GOSUB L4700
8360 'K=0;GOSUB L6700;F=BN;GOSUB L500;F=FU
8370 'IF RN>0 THEN UI=UI-1;GOSUB L5700;GOTO L829
0;FI
8390 'US=US+1;UV(US)=P-PB;IF US=1 THEN UW(UX)=E;
FI
8400 'GOTO L7950
8500 'L8500;I=US;IF US=0 THEN UX=UX-1;RETURN;FI
8520 'W=UV(I);W1=I+C1;IF W1=(W1/2)*2 THEN GOTO L
8550;FI
8530 'L8530;I=I-1;IF I=0 THEN GOTO L8600;FI
8540 'IF W>UV(I) THEN W=UV(I);FI
8550 'L8550;I=I-1;IF I=0 THEN GOTO L8600;FI
8560 'IF W<UV(I) THEN W=UV(I);FI
8570 'GOTO L8530
8600 'L8600;FOR UI=1 TO US;GOSUB L5700;NEXT
8610 'UZ(UX)=W;RETURN
8710 'L8700;UX=0;C1=1;FU=23;REPEAT
8720 'WU=B(FU);IF WU>14 THEN GOTO L8760;FI
8725 'IF (WU=10)*(FU<87) THEN GOTO L8760;FI
8730 'F=FU;GOSUB L500;IF RN=0 THEN GOTO L8760;FI
8740 'UX=UX+1;IF UX>20 THEN RETURN;FI
8745 'UY(UX)=FU
8750 'GOSUB L7900;GOSUB L8500
8760 'L8760;FU=FU+1;UNTIL FU>119;RETURN
8800 'L8800;UX=0;C1=0
8810 'FU=23;REPEAT
8820 'WU=B(FU);IF (WU>14)*(WU<>25) THEN GOTO L88
40;FI
8830 'IF (WU=10)*(FU<54) THEN GOTO L8840;FI
8835 'GOTO L8880
8840 'L8840;F=FU;GOSUB L500;IF QN=0 THEN GOTO L8
880;FI
8850 'UX=UX+1;IF UX>20 THEN RETURN;FI
8870 'UY(UX)=FU;GOSUB L7900;GOSUB L8500
8880 'L8880;FU=FU+1;UNTIL FU>119;RETURN
8920 'L8900;F=UF;GOSUB L500;IF RN=0 THEN FR=-1;R
ETURN;FI
8930 'UX=1;C1=1;FU=UF
8940 'GOSUB L7900;GOSUB L8500;IF UX=0 THEN FR=-1
;RETURN;FI
8950 'IF UD>=UZ(1) THEN FR=-1;RETURN;FI
8960 'HI=0;F=UF;E=UW(1);NA=0
8970 'GOSUB L7700;IF OT=-1 THEN OT=0;FR=-1;RETUR
N;FI
8980 'FR=0;RETURN
9000 'L9000;UE=UY(UX);UD=UZ(UX);UF=UW(UX)
9010 'IF B(UE)=10 THEN FR=0;RETURN;FI
9020 'E=UE;GOSUB L900;II=1
9030 'L9030;BW=S(II);WK=B(BW)
9040 'IF 14<WK THEN GOTO L9110;FI
9070 'E=UE;NA=0;HI=0;F=BW
9090 'GOSUB L9600;IF NG=-1 THEN NG=0;FR=-1;RETUR
N;FI
9110 'L9110;IF II<SN THEN II=II+1;GOTO L9030;FI
9120 'FR=0;RETURN
9220 'L9200;SM=0;FOR I4=0 TO 7;F=O1+DN(I4)
9230 'IF B(F)=25 THEN GOTO L9260;FI
9250 'GOSUB L500;SM=SM+QN
9260 'L9260;NEXT;RETURN
9305 'L9300;PA=P;FOR IP=1 TO TP
9310 'HI=-1;IF TG(IP)=-1 THEN HI=0;FI
9320 'E=TE(IP);F=TF(IP);NA=TA(IP);G=TG(IP)
9330 'GOSUB L4700
9340 'K=15;GOSUB L6700;F=BN;O1=BN;GOSUB L500
9350 'IF QN>0 THEN PZ(IP)=-1;GOTO L9400;FI
9360 'PZ(IP)=0
9370 'GOSUB L9200;PY(IP)=SM
9380 'GOSUB L8800
9382 'IF UX=0 THEN W=0;GOTO L9395;FI
9383 'IF UX=1 THEN GOTO L9390;FI
9385 'GOSUB L7600
9390 'L9390;W=UZ(UX);IF W>0 THEN W=0;FI
9395 'L9395;PX(IP)=W+P-PA+PV(IP)
9400 'L9400;GOSUB L5700;NEXT;RETURN
9500 'L9500;FR=0
9505 'IF TP>1 THEN GOTO L9520;FI
9510 'IF PZ(1)=-1 THEN FR=-1;RETURN;FI
9515 'IW=1;RETURN
9520 'L9520;IW=1
9525 'L9525;IF PZ(IW)=0 THEN GOTO L9540;FI
9530 'IW=IW+1;IF IW>TP THEN FR=-1;RETURN;FI
9535 'GOTO L9525
9540 'L9540;IF IW=TP THEN RETURN;FI
9545 'IP=IW+1
9550 'L9550;IF IP>TP THEN RETURN;FI
9555 'IF PZ(IP)=-1 THEN GOTO L9570;FI
9560 'IF PX(IP)>PX(IW) THEN IW=IP;GOTO L9570;FI
9565 'IF (PX(IP)=PX(IW))*(PY(IP)<PY(IW)) THEN IW
=IP;FI
9570 'L9570;IP=IP+1
9575 'GOTO L9550
9600 'L9600;TP=TP+1
9610 'TG(TP)=-1;IF HI=-1 THEN TG(TP)=G;FI
9620 'TE(TP)=E;TF(TP)=F;TA(TP)=NA
9630 'IF TP>19 THEN NG=-1;RETURN;FI
9640 'NG=0;RETURN
10000 'LINE;HLT[];CPOKE ADDR,$15,%6,%5,%1#,%2#,%
3#,%4#,%7;SUB[];RETURN
10100 'HLT;CODE $B6,$FD05#,$2BFB#;POKE $FD05,$80
;CODE $B6,$FD05#,$2AFB#;ADDR=$FC82;RETURN
10200 'SUB;POKE $FD05,0;RETURN
15000 'L15000;A=PEEK($6260);IF A=1 THEN GOSUB L5
700;FI;IF A=2 THEN GOSUB L4600;FI;IF A=3 THEN GO
SUB L5600;FI;IF A=4 THEN GOSUB L4700;FI;IF A=5 T
HEN GOSUB L6700;FI;IF A=6 THEN GOSUB L500;FI;IF
A=7 THEN GOSUB L6800;FI;IF A=8 THEN GOSUB L7000;
FI
15010 'IF A=9 THEN GOSUB L7200;FI;IF A=10 THEN G
OSUB L8700;FI;IF A=11 THEN GOSUB L7600;FI;IF A=1
2 THEN GOSUB L7700;FI;IF A=13 THEN GOSUB L9600;F
I;IF A=14 THEN GOSUB L8800;FI;IF A=15 THEN GOSUB
 L9000;FI;IF A=16 THEN GOSUB L8900;FI
15020 'IF A=17 THEN GOSUB L6300;FI;IF A=18 THEN
GOSUB L5900;FI;IF A=19 THEN GOSUB L9300;FI;IF A=
20 THEN GOSUB L9500;FI;IF A=21 THEN GOSUB L5500;
FI;IF A=22 THEN GOSUB L6200;FI;CODE $35,$FF;END
```

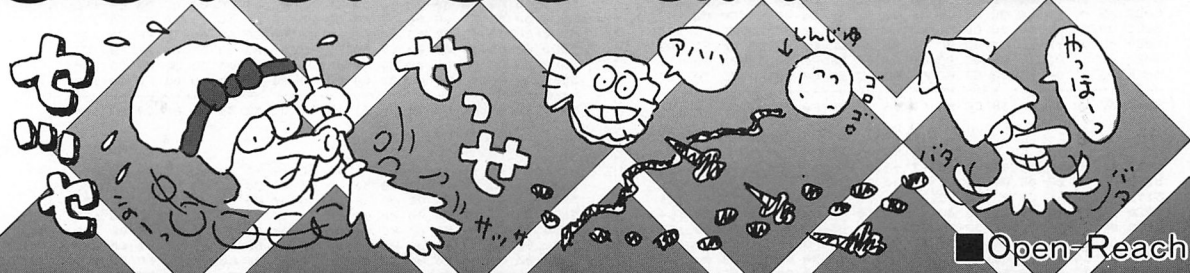
リスト3　データ・ダンプリスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1200 06 05 04 03 00 03 04 05 06 0A 02 0A 0A 0A 0A 0A :62
1210 01 0A 07 07 07 07 07 07 07 07 07 00 00 00 00 00 :4A
1220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
1240 6F 75 68 69 6B 61 6B 69 67 69 6B 65 6B 79 66 75 :B4
1250 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 :08
1260 00 00 00 02 00 00 00 02 00 02 00 00 00 02 00 00 :08
1270 00 02 00 00 00 02 00 00 00 00 00 02 00 00 00 02 :08
1280 00 01 00 01 00 01 00 01 00 01 00 00 00 01 00 00 :06
1290 00 01 00 01 00 01 00 00 00 00 00 01 00 00 00 01 :05
12A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
12B0 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 :02
12C0 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 :01
12D0 00 01 00 02 00 01 00 02 00 02 00 01 00 02 00 01 :0C
12E0 00 02 00 01 00 02 00 01 00 01 00 02 00 01 00 02 :0C
12F0 FF F4 FF F5 FF F6 FF FF 00 01 00 0A 00 0B 00 0C :FC
---------------------------------------------------------
Sum: 75 80 72 73 71 69 75 7B 74 82 74 80 75 95 70 92 :9A
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1300 FF E9 FF EA 00 00 00 00 00 00 00 00 00 00 00 00 :D1
1310 33 33 33 34 32 32 33 33 34 31 33 32 37 31 36 32 :31
1320 33 31 32 32 36 31 35 32 38 33 38 34 38 34 38 35 :46
1330 38 35 38 36 33 33 34 32 32 32 33 33 35 31 34 31 :3C
1340 34 33 34 34 35 32 34 33 34 31 33 31 33 31 32 32 :2E
1350 35 33 35 34 34 32 36 34 00 00 00 00 00 00 00 00 :A1
1360 00 64 00 16 00 14 00 10 00 0E 00 0A 00 08 00 02 :C0
1370 00 18 00 16 00 00 00 0F 00 0D 00 0C 00 09 FF 9C :FA
1380 FF EA FF EC FF F0 FF F2 FF F6 FF F8 FF FE FF E8 :84
1390 FF EA 00 00 FF F1 FF F3 FF F4 FF F7 00 00 00 00 :B4
13A0 00 00 00 02 00 02 00 00 00 01 00 03 00 04 00 07 :13
13B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
13C0 FF FE FF FE 00 00 FF FF FF FD FF FC FF F9 6D 61 :B5
---------------------------------------------------------
Sum: 03 36 03 06 02 F1 03 01 CF CA CE CE D5 D3 3F B8 :0D

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
```

リスト3　データ・ダンプリスト

```
5000 03 FF FF F0 00 00 C0 00 00 00 C0 00 00 00 C0 00 :31
5010 00 00 C0 00 00 00 C0 00 00 FF FF C0 00 00 C0 00 :FE
5020 00 00 C0 00 00 00 C0 C0 00 00 C0 30 00 00 C0 00 :F0
5030 0F FF FF FC 00 FF FC 00 00 03 0C 3C 00 33 0C FC :8A
5040 00 F3 0C 00 0F C3 03 00 00 C3 00 FC 0F FF FC 00 :9D
5050 00 C3 0C 3C 00 C3 0C FC 00 C3 0C 00 03 03 03 00 :AE
5060 0C 03 00 FC 00 03 00 00 00 0F FC 00 00 F0 30 00 :39
5070 0F 00 C0 00 00 FF FF C0 00 C0 C0 C0 00 C0 C0 C0 :0D
5080 00 FF FF C0 00 C0 C0 C0 00 C0 C0 C0 00 FF FF C0 :FC
5090 00 C0 00 C0 00 C0 00 C0 03 00 03 C0 00 00 C0 00 :86
50A0 00 03 30 00 00 0C 0C 00 00 30 03 00 00 CF FC C0 :09
50B0 03 00 C0 30 0C 00 C0 0C 00 FF FF C0 00 00 C0 00 :49
50C0 00 00 C0 00 00 C0 C0 C0 00 30 C3 00 0F FF FF FC :FC
50D0 00 0C 0F FC 00 33 0C 0C 00 C0 CC 0C 03 3F CF FC :07
50E0 0C 03 0C 0C 00 03 0C 0C 03 FF CF FC 00 03 0C 00 :1E
50F0 00 03 0F 0C 00 C3 0C F0 00 33 0C C0 00 FF CC 30 :D7
------------------------------------------------------
Sum: 3C 8B 2F E8 1B CC BA D0 06 68 82 F0 24 F3 5C 64 :06


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5100 0F 00 0F 0C 00 30 0C 00 00 30 0C 00 00 30 0C 00 :DE
5110 00 30 FF C0 0F FC 0C 00 00 30 0C 00 00 F3 FF F0 :24
5120 03 3C 0C 00 03 33 0C 00 0C 30 FF C0 00 30 0C 00 :C4
5130 00 30 0C 00 00 33 FF FC 00 00 03 C0 00 3F FC 00 :68
5140 00 00 C0 00 00 00 C0 00 03 FF FF F0 00 0C CC 00 :49
5150 00 30 C3 00 00 C0 C0 C0 03 3F FF 30 00 30 03 00 :D7
5160 00 3F FF 00 00 30 03 00 00 3F FF 00 00 00 C0 00 :6F
5170 00 00 C0 00 00 30 FF C0 00 30 C0 00 00 30 C0 00 :8F
5180 0F FF FF FC 00 00 C0 00 00 30 C3 00 00 30 C3 00 :AF
5190 00 C0 C3 C0 03 00 3C 30 00 0F C0 00 00 F0 00 00 :71
51A0 00 00 C0 00 03 FF FF F0 00 0C 0C 00 00 0C 0C 00 :E1
51B0 0F FF FF FC 00 00 00 00 00 FF FF C0 00 C0 C0 C0 :07
51C0 00 FF FF C0 00 C0 C0 C0 00 FF FF C0 00 00 C0 0C :88
51D0 00 00 FF FC 03 FF FF F0 03 00 C0 00 03 FF FF F0 :A0
51E0 03 00 C0 00 03 FF FF F0 03 00 C0 00 03 FF FF FC :74
51F0 00 00 00 0C 0C 30 C3 0C 0C 30 C3 0C 0C 30 C3 0C :2D
------------------------------------------------------
Sum: 33 C8 A7 4C 2A 9F 21 48 24 B6 A7 2C 12 18 72 B4 :1D


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5200 00 00 00 30 00 00 03 C0 00 00 00 00 00 00 00 00 :F3
5210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5230 00 00 00 00 00 00 00 00 00 00 00 00 00 30 0C 30 :6C
5240 03 FF 0C 30 00 30 00 30 00 C0 00 C0 00 0F 00 00 :2D
5250 00 30 CF FC 00 C0 0F FC 03 3F CF FC 0C 03 0C 00 :EE
5260 00 FF CF 0C 00 03 0C F0 00 33 0C 30 03 FF CF 0C :25
5270 00 C0 0C 30 0F FC 0C 30 00 C0 00 30 03 00 00 C0 :F6
5280 00 30 0C 00 00 30 0C 00 0F FC FF C0 00 30 0C 00 :7E
5290 00 F3 FF F0 03 30 0C 00 0C 30 FF C0 00 30 0C 00 :58
52A0 00 33 FF FC 00 30 0C 30 03 FF 0C 30 00 30 00 30 :38
52B0 00 C0 00 C0 00 00 00 00 00 FF FF 00 00 00 C0 00 :3E
52C0 03 FF FF F0 00 30 C3 00 00 C0 C0 C0 03 3F FF 30 :95
52D0 00 3F FF 00 00 3F FF 00 00 00 00 00 00 00 00 00 :7C
52E0 00 03 00 00 00 03 00 00 00 03 00 00 00 03 0F 00 :1B
52F0 00 03 F0 00 00 3C 00 00 00 C0 00 00 00 C0 00 00 :AF
------------------------------------------------------
Sum: 06 48 AE 34 12 2D 10 3C 21 9F A4 8C 15 D3 CD 5C :BC


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5300 00 C0 00 00 00 3F FF C0 00 00 00 00 0F FF FF FC :C7
5310 00 00 C0 00 03 00 C0 00 00 C0 C0 00 00 00 C0 00 :C3
5320 00 00 C0 00 00 FF FF C0 00 00 C0 00 00 00 C0 00 :FE
5330 00 00 C0 00 00 00 C0 00 00 00 C0 00 03 FF FF F0 :31
5340 00 00 30 00 0F C0 30 0C 00 30 30 30 00 0C 30 C0 :C7
5350 0F CC 30 C0 0F 0C 30 C0 00 0F FF FC 0F C0 30 C0 :9F
5360 00 30 30 FC 00 0C 33 C0 0F CC 33 00 0F 0C 30 00 :B4
5370 00 0F FF C0 00 F0 00 30 00 C0 00 C0 00 C0 00 C0 :EE
5380 00 FF FF C0 00 C0 C0 C0 00 C0 C0 C0 00 FF FF C0 :FC
5390 00 C0 C0 C0 00 C0 C0 C0 00 FF FF C0 00 00 C0 3C :3A
53A0 00 03 03 C0 00 0F FC 00 0F FF FF FC 00 30 C3 00 :CD
53B0 00 C0 C0 C0 00 00 C0 00 00 00 C0 00 00 FF FF C0 :7E
53C0 0C 00 C0 0C 03 00 C0 30 00 CF FC C0 00 30 03 00 :89
53D0 00 0C 0C 00 00 03 30 00 00 00 C0 00 0C 3C 00 3C :8F
53E0 03 0C FF C0 00 CC 33 00 03 CC 30 C0 0C 3C 30 00 :04
53F0 00 0C 30 00 0F FC FF F0 0C 0C 30 00 0C 0C 30 0C :D2
------------------------------------------------------
Sum: 1E 71 4C 48 33 60 6F DC 2D F0 3C E8 54 78 F2 30 :30


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5400 0F FC FF 30 0C 0C C0 C0 0C 0C 33 00 0F FC 0C 00 :34
5410 0F FF F3 00 00 0C 03 00 00 0C 03 00 00 FF C3 0C :ED
5420 00 0C 33 30 00 0C 0F 30 03 FF F3 C0 00 0C 03 00 :7E
5430 00 0C 0F FC 00 FF C3 00 00 0C 03 00 00 0C 03 00 :F7
5440 00 0C 03 00 00 3F FF 00 00 30 03 00 00 3F FF 00 :BE
5450 00 30 03 00 03 3F FF 30 00 C0 C0 C0 00 30 C3 00 :D7
5460 00 0C CC 00 03 FF FF F0 00 00 C0 00 00 00 C0 00 :49
5470 00 0F FF 00 00 F0 00 00 00 00 03 C0 00 00 FC 00 :BD
5480 03 0F 00 30 00 F0 C0 C0 00 30 C3 00 00 30 C3 00 :98
5490 00 00 C0 00 0F FF FF FC 00 00 C3 00 00 00 C3 00 :4F
54A0 00 FF C3 00 00 00 C0 00 00 00 C0 00 0F FF C0 00 :10
54B0 0C 00 C0 00 00 FF FF C0 00 C0 C0 C0 00 FF FF C0 :88
54C0 00 C0 C0 C0 00 FF FF C0 00 00 00 00 0F FF FF FC :07
54D0 00 0C 0C 00 00 0C 0C 00 03 FF FF F3 00 00 C0 00 :E4
54E0 00 F0 00 00 03 00 00 00 0C 30 C3 0C 0C 30 C3 0C :09
54F0 0C 30 C3 0C 0C 00 00 00 0F FF FF F0 00 00 C0 30 :04
------------------------------------------------------
Sum: 39 64 D7 58 30 89 1B 4C 2D 31 79 EF 39 DF DA 04 :A8


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
```

```
5500 03 FF FF F0 00 00 C0 30 03 FF FF F0 00 00 C0 30 :C2
5510 03 FF FF F0 00 00 00 00 00 00 00 00 00 00 00 00 :F1
5520 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5530 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5540 00 00 00 00 00 00 00 00 0C 3C FF F0 03 0C 33 00 :79
5550 03 CC 30 00 0C 3C FF C0 00 0C 30 0C 0F FC FF 30 :88
5560 0F FC 00 C0 0F FC C3 00 00 00 3C 00 00 C0 00 C0 :55
5570 03 00 03 00 03 0C 3F F0 03 0C 03 00 0F FF F3 00 :57
5580 00 0C 03 00 00 FF C3 0C 00 0C 03 30 03 FF F3 C0 :D1
5590 00 0C 03 00 00 FF CF FC 00 0C 03 00 00 0C 03 00 :F7
55A0 00 C0 00 30 03 00 00 C0 03 0C 0F FC 03 0C 00 C0 :9C
55B0 00 3F FF 00 00 3F FF 00 03 3F FF 30 00 C0 C0 C0 :2D
55C0 00 30 C3 00 03 FF FF F0 00 00 C0 00 00 3F FF C0 :A2
55D0 00 00 00 00 00 C0 00 C0 03 00 03 00 03 0C 3F F0 :C4
55E0 03 0C 03 00 00 00 00 00 00 FF FF 00 00 00 00 C0 :D0
55F0 00 00 00 C0 00 00 00 C0 00 00 0F 00 00 03 F0 00 :82
------------------------------------------------------
Sum: 1E 19 FC 90 24 40 51 18 1B B5 52 48 2A EC C9 D0 :A9


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5600 00 3C 30 00 00 00 30 00 00 00 30 00 00 00 30 00 :FC
5610 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5620 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0F F0 :FF
5630 0F FF FF FC 03 FC 00 00 00 00 00 00 00 00 00 00 :08
5640 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5650 00 00 00 00 00 00 00 00 00 0F 00 00 FF FF C0 :CD
5660 00 3F 00 00 00 00 00 00 00 00 3F F0 0F FF FF FC :77
5670 03 FC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :FF
5680 00 00 00 00 00 00 00 00 00 00 3F 00 00 FF FF C0 :FD
5690 00 3F 00 00 00 00 0C 00 00 3F FF 00 00 0F 00 00 :98
56A0 00 00 0F F0 0F FF FF FC 03 FC 00 00 00 00 00 00 :07
56B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 30 :30
56C0 0F FF FF FC 0F 03 0C 3C 0F 0F 0C 3C 0F 3C 0F FC :1F
56D0 0F C0 00 3C 0F FF FF FC 0F 00 00 3C 00 00 00 00 :5F
56E0 00 00 00 00 00 00 00 00 00 00 03 C0 00 FF FF F0 :B1
56F0 00 00 30 00 00 00 30 00 00 00 C0 00 00 3F FF C0 :1E
------------------------------------------------------
Sum: 30 74 6D 24 30 FD 76 34 21 4A 8B 28 1E 86 49 48 :5F


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5700 00 00 C0 C0 00 03 00 C0 00 03 00 C0 03 FF FF FC :03
5710 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C0 00 :C0
5720 00 00 C0 C0 03 FF FF F0 00 00 00 00 00 0F 0C 00 :8C
5730 00 0F 03 00 00 3C 03 C0 00 30 03 C0 00 C0 03 C0 :87
5740 03 00 00 C0 00 00 00 00 00 00 00 00 00 00 00 00 :C3
5750 00 00 00 00 00 03 00 00 00 03 00 C0 00 03 03 F0 :BC
5760 00 0F FC 00 03 F3 00 00 00 03 00 00 00 03 00 00 :07
5770 00 03 00 30 00 03 FF F0 00 00 00 00 00 00 00 00 :25
5780 00 00 00 00 00 00 0C 00 00 3F FF 00 00 00 0C 00 :56
5790 00 0F 0C 00 00 0F 0C 00 00 0F 0C 00 00 3C 03 00 :90
57A0 00 3C 03 00 00 30 00 C0 00 C0 00 30 00 00 00 00 :1F
57B0 00 00 00 00 00 00 00 00 00 03 C0 00 00 03 00 00 :C6
57C0 00 03 03 00 03 FF FF C0 00 03 03 00 00 03 03 00 :D3
57D0 00 0C 03 00 00 0C 03 00 00 30 03 00 00 C0 03 0C :20
57E0 03 00 03 FC 00 00 00 00 00 00 00 00 00 00 00 00 :02
57F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0F F0 :FF
------------------------------------------------------
Sum: 06 7B 97 6C 09 81 1B E0 00 7D D4 70 03 D6 F5 A8 :40


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5800 0F FF FF FC 03 FC 00 00 00 00 00 00 00 00 00 00 :08
5810 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
5820 00 00 00 00 00 00 00 00 00 00 0F F0 0F FF FF FC :08
5830 03 FF 00 00 00 00 00 00 00 00 3F 00 00 FF FF C0 :FF
5840 00 3C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :3C
5850 00 00 00 00 00 00 00 00 00 00 0F F0 0F FF FF FC :08
5860 03 FC 00 00 00 00 3C 00 00 3F FF 00 00 0C 00 00 :85
5870 00 00 3F 00 00 FF FF C0 00 3F 00 00 00 00 00 00 :3C
5880 00 00 00 00 00 00 00 00 00 00 00 00 0F 00 00 3C :4B
5890 0F FF FF FC 0F 00 00 3C 0F 00 00 FC 0F FC 0F 3C :B5
58A0 0F 0C 3C 3C 0F 0C 30 3C 0F FF FF FC 03 00 00 00 :26
58B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
58C0 0F FF FF F0 00 C0 30 00 00 C0 30 00 00 C0 C0 00 :5D
58D0 00 FF FF 00 00 00 C0 00 00 03 00 00 00 03 00 00 :C4
58E0 03 FF FF C0 00 F0 00 00 00 00 00 00 00 00 00 00 :B1
58F0 00 00 00 00 00 C0 00 30 00 F0 00 C0 00 F0 03 00 :93
------------------------------------------------------
Sum: 45 3E 76 E4 21 77 5B 68 1E 30 8B 98 3F B8 CF 30 :9F


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
5900 00 F0 0F 00 00 30 3C 00 00 0C 3C 00 00 00 00 00 :B3
5910 03 FF FF F0 00 C0 C0 00 00 00 C0 00 00 00 00 00 :31
5920 00 00 00 00 00 00 00 00 03 FF F0 00 03 00 30 00 :25
5930 00 00 30 00 00 00 30 00 00 00 33 F0 00 0F FC 00 :8E
5940 03 F0 30 00 00 C0 30 00 00 00 30 00 00 00 00 00 :43
5950 00 00 00 00 00 00 00 00 00 00 00 00 03 00 00 C0 :C3
5960 00 C0 03 00 00 30 0F 00 00 30 0F 00 00 0C 3C 00 :89
5970 00 0C 3C 00 00 0C 3C 00 00 0C 00 00 00 3F FF 00 :DA
5980 00 0C 00 00 00 00 00 00 00 00 00 00 0F F0 00 30 :3B
5990 0C 30 00 C0 00 30 03 00 00 30 0C 00 00 30 0C 00 :A7
59A0 00 30 30 00 00 30 30 00 00 FF FF F0 00 30 30 00 :0E
59B0 00 00 30 00 00 00 F0 00 00 00 00 00 00 00 00 00 :20
59C0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
59D0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
59E0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
59F0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 :50
------------------------------------------------------
Sum: A6 AB A1 44 94 E0 5E 94 97 0A FD 74 A9 3E 37 84 :50
```

あなたはタコです．イカではありません．きれい好きのタコくんは毎日お庭を掃除しています．青いボールに当たると気が狂うので，なかなか掃除ができません．ときどき，たちの悪いイカやホタテがやってきて，せっかくきれいにした庭を汚してしまったり，追いかけてきたりします．この際，イカを食べるべし．ホタテをなめるなよ!?

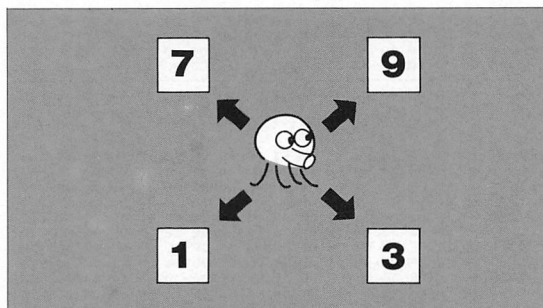
遊び方

タコを[1][3][7][9]キーを使って左下，右下，左上，右上に移動させて，庭の色を水色に塗り変えてください．全部塗り変えると一面クリアです．

青いボールに当ると死んでしまいます．ときどきイカがやってきて庭の色を青に戻してしまいますが，イカを食べるとBONUS 500点なので，早いうちに食べるべきでしょう．さらに，しつこいホタテがやってきてタコを追い駆けてくるので注意しましょう．

5面クリアすると次のレベルに進みます．レベルは1～5まであり，段々と難しくなります．庭の色の変化は次の通りです．

●レベル1　　青──→水色
●レベル2　　青──→ピンク──→水色
●レベル3　　青──→水色──→青──→…
●レベル4　　青──→ピンク──→水色──→青──→…
●レベル5　　青──→ピンク──→水色──→緑──→青──→…

すべての面が水色にならないとクリアできません．レベル3以降では水色にしたところに再びいくと元にもどってしまいます．レベル3以降をクリアするにはパズル的なセンスが要求されるようです．タコは5000点ごとに一匹増えますが，10匹以上には増えません．

高得点をあげるにはホタテの退治方法を見つけなければなりませんが，その方法は自分で見つけてください．ちなみに筆者のハイ・スコアは42,000点です．

なお，プログラムはすべてマシン語（$1400～$3CF9）で，スタート番地は$1700です．

図1　キー操作

◆参考文献

1）FM-8活用研究，工学社

ななめに動くのがムズカシイ．

ボールが直撃！

OCTOPUS GARDENマシン語ダンプ・リスト

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1400 34 7D 86 00 1F 8B 86 01 97 BF BD D0 72 35 FD FB :EA
1410 FA 25 06 6D 07 27 F6 A6 06 35 FD 34 7F C6 00 1F :2C
1420 9B C6 02 D7 BF BD D0 8E 35 FF CC 00 01 ED 04 AD :B3
1430 9F FB FA 35 FF 34 7F 86 00 1F 8B BD D8 07 25 0C :78
1440 A6 80 81 00 26 FA 86 0D A7 82 35 FF 7E 94 8D 00 :56
1450 17 02 8D 30 8D 02 69 A6 03 E6 88 16 C0 11 A2 88 :F6
1460 1A 34 06 43 25 01 30 8D 02 8E 10 32 62 35 90 12 :85
1470 F5 FD 0F BD D0 8E B7 FD 0F 39 00 00 00 00 00 00 :18
1480 16 FF 98 16 FF 7A 16 FF AC 39 0C 35 10 20 02 8D :36
1490 EF A6 80 26 FA 6E 84 86 0D 8D E5 86 0A 20 E1 36 :F3
14A0 06 A6 43 3D 34 06 EC 41 3D EB E4 E7 E4 A6 C4 E6 :BA
14B0 43 3D EB E0 A6 E0 1E 89 33 44 39 8E 00 00 34 10 :FA
14C0 34 10 34 10 4D 2A 04 8D 40 63 61 6C 60 2B 40 58 :23
14D0 49 2A F8 44 56 ED 64 EC C1 2A 04 8D 2C 63 61 6A :18
14E0 60 2B 1E 58 49 2A F8 44 56 A3 64 20 06 58 49 24 :F8
14F0 F8 E3 64 1C FE 2B 02 1A 01 69 63 69 62 6A 60 2A :2C
-----------------------------------------------------------
Sum: 57 E6 9F CA 49 68 A7 1E 0E CF 18 BA 5C FF 0A 36 :66

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1500 EC EC 62 6D 61 32 66 2A 05 43 53 C3 00 01 39 32 :94
1510 66 CC 7F FF 39 4D 2B F1 39 A3 C1 2F 2C CC 00 00 :16
1520 39 A3 C1 2E 24 CC 00 00 39 A3 C1 2D 1C CC 00 00 :6D
1530 39 A3 C1 2C 14 CC 00 00 39 A3 C1 27 0C CC 00 00 :45
1540 39 A3 C1 26 04 CC 00 00 39 CC 00 01 39 83 00 00 :55
1550 27 F7 CC 00 00 39 17 FF 2A 1F 89 4F 39 8D 15 16 :4B
1560 FF A7 17 FE EB 17 FF 1E A6 80 81 24 27 2B 81 2D :A5
1570 27 EB 30 1F CC 00 00 34 06 A6 84 82 30 2B 18 81 :07
1580 09 22 14 A7 84 EC 60 58 49 58 49 E3 60 58 49 EB :C7
1590 80 89 00 ED 60 20 E2 35 86 CC 00 00 34 06 A6 84 :43
15A0 82 30 2B F3 81 09 23 06 82 07 81 0F 22 E9 A7 84 :D2
15B0 EC 60 58 49 58 49 58 49 58 49 EB 80 ED 60 20 DE :86
15C0 8E 00 00 36 10 36 10 36 10 36 10 4D 2A 05 63 40 :C5
15D0 17 00 D7 58 49 ED 46 8E 00 0F C6 05 1C FE A6 C5 :AF
15E0 49 81 09 22 04 1C FE 20 04 82 0A 1A 01 A7 C5 5A :A4
15F0 26 EC 68 47 69 46 24 02 6C 45 30 1F 26 DC C6 01 :5F
-----------------------------------------------------------
Sum: 55 D2 16 D0 10 16 DC 2E E8 BD E9 39 2D F8 31 27 :81

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1600 A6 C5 26 05 5C C1 05 26 F7 39 8D B4 34 04 EB 49 :BB
1610 C2 06 23 02 8D 20 35 04 6D 40 2A 05 86 2D 17 FE :77
1620 5F A6 C5 8B 30 17 FE 58 5C C1 06 26 F4 33 4A 39 :E5
1630 33 5E 8D 8C 20 E2 86 20 17 FE 45 5A 26 FA 39 34 :93
1640 04 8D 04 35 04 1F 98 1F 89 44 44 44 44 8D 04 1F :ED
1650 98 84 0F 8B 30 81 39 23 02 8B 07 16 FE 22 1F 98 :44
1660 20 F9 33 5C A6 45 E6 47 3D ED 42 EC 45 3D EB 42 :C7
1670 89 00 ED 41 A6 44 E6 47 3D E3 41 ED 41 A6 44 E6 :2D
1680 46 3D EB 41 89 00 ED C4 39 34 06 EC 8C 1A 8E 3D :B9
1690 09 36 16 8D CD EC 42 33 48 C3 00 03 ED 8C 09 35 :D5
16A0 10 36 16 8D BD 33 48 39 1C 2C 6A 49 16 FE 5A 20 :E3
16B0 7C, 4F 34 3B 8C 00 00 00 00 00 00 00 00 00 00 00 :C6
16C0 43 6F 70 79 72 69 67 68 74 20 28 63 29 20 31 39 :17
16D0 38 32 20 62 79 20 4E 2E 54 73 75 64 61 00 8E 10 :A0
16E0 30 8C DD C6 1D 4F AB 80 5A 26 FB 81 19 26 01 39 :6B
16F0 8E 80 00 4F A7 82 2D FC 2E FA BD 1F 34 35 76 12 :A4
-----------------------------------------------------------
Sum: 53 7E 86 01 07 7C 5F B4 C9 AD 95 0B 02 0F FE B9 :CC

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1700 34 7F 86 6F 1F 8B 5F 1F 03 10 DF FE 30 8D 2E B3 :5E
1710 9F 00 30 8D 26 DB 9F 02 30 8D 26 6F 9F 04 30 8D :B0
1720 26 53 9F 06 30 8D 26 37 9F 08 30 8D 26 1B 9F 0A :86
1730 30 8D 25 FF 9F 0C 30 8D 25 E3 9F 0E 30 8D 25 CB :AB
1740 9F 10 30 8D 25 B3 9F 12 17 0B EC 17 16 58 17 17 :B6
1750 19 CC 00 00 DD 14 30 8D 1D 67 1F 10 36 06 CC C5 :13
1760 00 34 06 37 10 34 10 17 18 3F 32 64 30 8D 1C 91 :33
1770 1F 10 36 06 CC C5 00 C3 00 C0 34 06 37 10 34 10 :44
1780 17 18 26 32 64 30 8D 1D F8 1F 10 36 06 CC CE 40 :02
1790 34 06 37 10 34 10 17 18 10 32 64 30 8D 1B A2 1F :33
17A0 10 36 06 CC C6 80 34 06 37 10 34 10 17 17 FA 32 :7D
17B0 64 30 8D 1E 8C 1F 10 36 06 CC CB C0 34 06 37 10 :0E
17C0 34 10 17 17 E4 32 64 30 8D 1F 36 1F 10 36 06 CC :35
17D0 CC 80 34 06 37 10 34 10 17 17 CE 32 64 30 8D 1B :7B
17E0 30 1F 10 36 06 CC C7 40 34 06 37 10 34 10 17 17 :61
17F0 89 32 64 CC 00 00 DD 16 16 00 1C CC CF 00 D3 16 :94
-----------------------------------------------------------
Sum: 78 E4 95 16 FD AC 57 65 76 62 0F FC 2D AE 73 47 :E4

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1800 36 06 CC 00 00 34 06 37 10 34 10 17 14 A9 32 64 :37
1810 DC 16 C3 00 01 DD 16 83 00 BF 2F DF CC 00 00 DD :A2
1820 16 16 00 16 DC 16 58 49 9E 04 30 8B 36 10 CC 00 :44
1830 00 ED D1 DC 16 C3 00 01 DD 16 83 00 32 2F E5 CC :FC
1840 00 00 DD 16 16 00 16 DC 16 58 49 9E 00 30 8B 36 :41
1850 10 CC 00 20 ED D1 DC 16 C3 00 01 DD 16 83 03 E8 :D1
1860 2F E5 CC 00 00 DD 16 16 00 16 DC 16 58 49 9E 02 :32
1870 30 8B 36 10 CC 00 00 ED D1 DC 16 C3 00 01 DD 16 :34
1880 83 03 E8 2F E5 17 1F 39 CC 00 00 DD 18 CC 00 05 :83
1890 DD 1A CC 00 05 DD 1C CC 00 01 DD 1E CC 00 00 DD :32
18A0 20 CC 00 01 DD 22 CC 00 00 DD 24 17 03 30 CC 00 :CF
18B0 07 DD 26 CC 00 1E 36 06 CC 00 0A 34 06 37 10 34 :BB
18C0 10 17 15 73 32 64 17 FB C2 50 75 73 68 00 CC 00 :85
18D0 1E 36 06 CC 00 0B 34 06 37 10 34 10 17 15 58 32 :AC
18E0 64 17 FB A7 52 65 74 75 72 6E 20 4B 65 79 00 CC :B2
18F0 00 01 DD 16 DC 1C 36 06 DC 16 16 00 10 DC 16 34 :66
-----------------------------------------------------------
Sum: B0 86 0C 30 E9 BC AE 80 14 19 18 E9 97 82 02 8B :19

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1900 06 17 06 4E 32 62 DC 16 C3 00 01 DD 16 A3 C4 2F :44
1910 EC 33 42 CC 00 01 36 06 CC 00 01 34 06 37 10 34 :EC
1920 10 17 05 FE 32 64 DC 2A DD 28 CC 00 1E DD 2C CC :8A
1930 00 01 DD 2E CC 00 00 DD 30 CC 00 00 DD 32 17 06 :DD
1940 92 CC 00 01 DD 16 16 00 07 DC 16 C3 00 01 DD 16 :18
1950 83 3A 98 2F F4 17 13 EA 83 00 0D 26 E1 CC 00 02 :F1
1960 DD 1C CC 00 01 34 06 17 05 E8 32 62 CC 00 02 34 :9A
1970 06 17 05 DE 32 62 CC 00 01 36 06 CC 00 01 34 06 :A4
1980 37 10 34 10 17 05 9B 32 64 DC 2A DD 28 DC 34 DD :D0
1990 2C 17 02 4A DC 18 DD 36 17 0C 2F 17 0F 90 CC 00 :6A
19A0 07 DD 26 CC 00 06 36 06 CC 00 00 34 06 37 10 34 :99
19B0 10 17 14 83 32 64 CC 00 05 36 06 DC 18 17 FC 4A :B2
19C0 17 FA C8 30 00 CC 00 01 DD 16 DC 1C 36 06 DC 16 :EF
19D0 16 00 22 DC 16 34 06 17 0A F2 32 62 DC 16 34 06 :37
19E0 17 06 15 32 62 DC 16 34 06 17 0A E0 32 62 DC 16 :79
19F0 C3 00 01 DD 16 A3 C4 2F DA 33 42 17 0B CC DC 24 :8A
-----------------------------------------------------------
Sum: 7B B6 03 18 E7 90 43 0D 3F 5E E2 A1 68 BB FE 38 :8C

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1A00 D3 18 93 36 DD 24 DC 24 83 01 F4 10 2F 00 41 DC :89
1A10 24 83 01 F4 DD 24 DC 1A 83 00 0A 10 2C 00 31 DC :69
1A20 1A C3 00 01 DD 1A CC 00 00 36 06 DC 1A 58 49 83 :F7
1A30 00 01 36 06 CC C5 00 34 06 37 10 34 10 37 10 34 :0E
1A40 10 17 16 37 32 66 CC 00 C8 34 06 17 10 C9 32 62 :5E
1A50 CC 00 01 DD 16 CC 13 88 36 06 DC 1E 36 06 CC 01 :66
1A60 F4 17 FA 3B A3 C1 53 43 C3 00 01 36 06 DC 16 16 :42
1A70 00 07 DC 16 C3 00 01 DD 16 A3 C4 2F F5 33 42 17 :C7
1A80 0E AC DC 32 83 00 1C 10 26 00 03 17 0F 64 DC 20 :26
1A90 83 00 00 10 27 FE FD DC 20 83 00 01 10 26 00 1D :88
1AA0 DC 28 36 06 DC 2C 36 06 CC CE 40 34 06 37 10 34 :13
1AB0 10 37 10 34 10 17 15 C3 32 66 17 0E D6 DC 20 83 :9C
1AC0 00 02 10 26 00 39 DC 28 36 06 DC 2C 34 06 37 10 :3A
1AD0 34 10 17 08 E3 32 64 17 0A 98 CC 00 01 36 06 CC :6A
1AE0 00 01 34 06 37 10 34 10 17 04 37 32 64 CC 00 13 :8D
1AF0 DD 28 CC 00 04 DD 2C CC 00 00 DD 30 17 12 43 17 :3A
-----------------------------------------------------------
Sum: 6F DA 00 46 C5 B3 BB EA 7E A4 D1 B2 71 24 AD F9 :8C

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1B00 0F 73 DC 1A 83 00 01 DD 1A CC 00 00 DD 20 CC 00 :88
1B10 00 36 06 DC 1A 58 49 C3 00 01 36 06 CC CF 00 34 :A2
1B20 06 37 10 34 10 37 10 34 10 17 15 4F 32 66 DC 1A :25
1B30 83 00 00 10 26 FE 5D CC 00 07 DD 26 CC 00 0C 36 :F8
1B40 06 CC 00 0A 34 06 37 10 34 10 17 12 EA 32 64 17 :61
1B50 F9 39 47 20 41 20 4D 20 45 20 20 4F 20 56 20 45 :16
1B60 20 52 00 CC 00 0F 36 06 CC 00 18 34 06 37 10 34 :22
1B70 10 17 12 C3 32 64 17 F9 12 72 65 70 6C 61 79 3F :80
1B80 28 59 2F 4E 29 00 DC 18 93 14 10 2F 00 04 DC 18 :F9
1B90 DD 14 17 11 AD DD 38 DC 38 36 06 CC 00 59 17 F9 :60
1BA0 98 36 06 DC 38 36 06 CC 00 79 17 F9 8C AA C0 EA :59
1BB0 C0 10 27 00 03 16 FC D0 DC 38 36 06 CC 00 4E 17 :5D
1BC0 F9 77 36 06 DC 38 36 06 CC 00 6E 17 F9 6B AA C0 :1B
1BD0 EA C0 10 27 00 05 10 DE FE 35 FF 16 FF B4 CC 00 :9B
1BE0 00 34 06 17 12 1C 32 62 CC 00 01 DD 16 16 00 3C :25
1BF0 CC 00 01 DD 3A DC 16 36 06 DC 3A 16 00 21 DC 16 :51
-----------------------------------------------------------
Sum: D3 6C 0B 4F B3 84 2C DB C4 99 E7 9A 89 D2 14 77 :9B

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1C00 36 06 DC 3A 36 06 CC 00 01 34 06 37 10 34 10 37 :57
1C10 10 34 10 17 01 13 32 66 DC 3A C3 00 01 DD 3A A3 :AB
1C20 C4 2F DB 33 42 DC 16 C3 00 01 DD 16 83 00 07 2F :A5
1C30 BF CC 00 07 DD 26 CC 00 00 36 06 CC 00 00 34 06 :A3
1C40 37 10 34 10 17 11 F0 32 64 17 F8 3F 73 63 6F 72 :3E
1C50 65 3A 00 CC 00 05 36 06 DC 18 17 F9 AD 17 F8 2B :97
1C60 30 00 CC 00 0D 36 06 CC 00 00 34 06 37 10 34 10 :D6
1C70 17 11 C4 32 64 17 F8 13 68 69 2D 73 63 6F 72 65 :BE
1C80 3A 00 CC 00 05 36 06 DC 14 17 F9 7E 17 F7 FC 30 :FF
1C90 00 CC 00 01 DD 16 DC 1A 83 00 01 36 06 DC 16 16 :7E
1CA0 00 27 CC 00 00 36 06 DC 16 58 49 C3 00 01 36 06 :C2
1CB0 CC C5 00 34 06 37 10 34 10 37 10 34 10 17 13 BB :C6
1CC0 32 66 DC 16 C3 00 01 DD 16 A3 C4 2F D5 33 42 CC :ED
1CD0 00 1D 36 06 CC 00 00 34 06 37 10 34 10 17 11 57 :69
1CE0 32 64 17 F7 A6 6C 76 3A 20 00 DC 22 17 F9 41 CC :A1
1CF0 00 01 DD 16 DC 1E 36 06 DC 16 16 00 25 CC 00 22 :45
-----------------------------------------------------------
Sum: 16 30 29 F7 D7 C1 A9 97 5A D3 35 FA 9C 04 81 39 :F4

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1D00 D3 16 36 06 CC 00 00 36 06 CC C7 40 34 06 37 10 :81
1D10 34 10 37 10 34 10 17 12 BF 32 66 DC 16 C3 00 01 :05
1D20 DD 16 A3 C4 2F D7 33 42 39 EC 62 36 06 EC 64 34 :1C
1D30 06 37 10 34 10 17 01 EA 32 64 EC 64 36 06 EC 62 :03
1D40 36 06 CC 00 07 17 F7 57 E3 C1 83 00 07 58 49 9E :E1
1D50 04 30 8B 36 10 EC 66 ED D1 DC 2A 36 06 DC 34 83 :EA
1D60 00 01 36 06 CC 00 E4 36 06 EC 66 34 06 37 10 34 :30
1D70 10 37 10 34 10 37 10 34 10 17 06 DA 32 68 DC 2A :BD
1D80 C3 00 01 36 06 DC 34 83 00 01 36 06 CC 00 E5 36 :B7
1D90 06 EC 66 34 06 37 10 34 10 37 10 34 10 37 10 34 :23
1DA0 10 17 06 B2 32 68 DC 2A 83 00 01 36 06 DC 34 36 :85
1DB0 06 CC 00 E4 36 06 EC 66 34 06 37 10 34 10 37 10 :50
1DC0 34 10 37 10 34 10 17 06 8D 32 68 DC 2A 36 06 DC :31
1DD0 34 36 06 CC 00 87 36 06 EC 66 34 06 37 10 34 10 :16
1DE0 37 10 34 10 37 10 34 10 17 06 6B 32 68 DC 2A C3 :01
1DF0 00 01 36 06 DC 34 36 06 CC 00 87 36 06 EC 66 34 :9E
-----------------------------------------------------------
Sum: B2 07 D1 70 ED 94 5F 8B 1D CA A0 C4 B0 BF 1A B9 :F2

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1E00 06 37 10 34 10 37 10 34 10 37 10 34 10 17 06 46 :0A
1E10 32 68 DC 2A C3 00 02 36 06 DC 34 36 06 CC 00 E5 :9E
1E20 36 06 EC 66 34 06 37 10 34 10 37 10 34 10 37 10 :25
1E30 34 10 17 06 21 32 68 DC 2A 83 00 01 36 06 DC 34 :F2
1E40 C3 00 01 36 06 CC 00 E6 36 06 EC 66 34 06 37 10 :C1
1E50 34 10 37 10 34 10 37 10 34 10 17 05 F9 32 68 DC :E5
1E60 2A 36 06 DC 34 C3 00 01 36 06 CC 00 87 36 06 EC :F1
```

OCTOPUS GARDEN

OCTOPUS GARDENマシン語ダンプ・リスト

```
1E70 66 34 06 37 10 34 10 37 10 34 10 37 10 34 10 17 :58
1E80 05 D4 32 68 DC 2A C3 00 01 36 06 DC 34 C3 00 01 :4D
1E90 36 06 CC 00 87 36 06 EC 66 34 06 37 10 34 10 37 :19
1EA0 10 34 10 37 10 34 10 17 05 AC 32 68 DC 2A C3 00 :0A
1EB0 02 36 06 DC 34 C3 00 01 36 06 CC 00 E7 36 06 EC :29
1EC0 66 34 06 37 10 34 10 37 10 34 10 37 10 34 10 17 :58
1ED0 05 84 32 68 DC 2A 36 06 DC 34 C3 00 02 36 06 CC :42
1EE0 00 E6 36 06 EC 66 34 06 37 10 34 10 37 10 34 10 :C4
1EF0 37 10 34 10 17 05 5F 32 68 DC 2A C3 00 01 36 06 :A6
--------------------------------------------------------
Sum: 18 21 E9 53 3C 62 AA FD 51 66 95 A2 94 6D 27 7B :4B

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
1F00 DC 34 C3 00 02 36 06 CC 00 E7 36 06 EC 66 34 06 :8C
1F10 37 10 34 10 37 10 34 10 37 10 34 10 17 05 37 32 :26
1F20 68 39 CC 00 11 36 06 EC 62 58 49 A3 C1 53 43 C3 :66
1F30 00 01 36 06 EC 64 58 49 58 49 E3 C1 DD 2A CC 00 :46
1F40 01 36 06 EC 62 36 06 CC 00 03 17 F5 52 E3 C1 DD :75
1F50 34 39 EC 62 58 49 9E 06 30 8B 36 10 CC 00 13 ED :CD
1F60 D1 EC 62 58 49 9E 08 30 8B 36 10 CC 00 1E 17 F7 :5F
1F70 18 40 50 82 00 58 49 83 00 0A ED D1 EC 62 58 49 :05
1F80 9E 0C 30 8B 36 10 CC 00 00 ED D1 EC 62 58 49 9E :C2
1F90 0E 30 8B 36 10 CC 00 00 ED D1 CC 00 0A 17 F6 E9 :65
1FA0 83 00 00 10 26 00 0F EC 62 58 49 9E 0E 30 8B 36 :54
1FB0 10 CC 00 01 ED D1 CC 00 32 17 F6 CD 83 00 00 10 :06
1FC0 26 00 0F EC 62 58 49 9E 0E 30 8B 36 10 CC 00 02 :9F
1FD0 ED D1 39 CC 00 01 DD 16 DC 1C 36 06 DC 16 16 00 :F3
1FE0 10 DC 16 34 06 17 00 10 32 62 DC 16 C3 00 01 DD :8A
1FF0 16 A3 C4 2F EC 33 42 39 EC 62 9E 0E 58 49 EC 8B :58
--------------------------------------------------------
Sum: 11 71 7A 2B E6 A5 9C 7F 35 A3 F7 D3 AF 15 8A 3C :F9

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2000 83 00 00 10 26 00 08 CC C6 80 DD 3C 16 00 1C EC :0A
2010 62 9E 0E 58 49 EC 8B 83 00 01 10 26 00 08 CC CB :7F
2020 C0 DD 3C 16 00 05 CC CC 80 DD 3C EC 62 9E 0C 58 :75
2030 49 EC 8B 83 00 02 10 26 00 9C DC 3C 83 CB C0 10 :4D
2040 26 00 5E EC 62 9E 06 58 49 EC 8B 36 06 EC 62 9E :B6
2050 08 58 49 EC 8B 34 06 37 10 34 10 17 04 4B 32 64 :E1
2060 DC 40 36 06 CC 00 07 17 F4 35 D3 42 83 00 07 9E :A8
2070 04 58 49 EC 8B DD 3E DC 3E 83 00 05 10 26 00 07 :16
2080 DC 32 83 00 01 DD 32 DC 40 36 06 DC 42 36 06 CC :1F
2090 00 01 34 06 37 10 34 10 37 10 34 10 17 FC 8A 32 :20
20A0 66 EC 62 58 49 9E 0C 30 8B 36 10 CC 00 03 ED D1 :8D
20B0 EC 62 9E 06 58 49 EC 8B 36 06 EC 62 9E 08 58 49 :DB
20C0 EC 8B 36 06 DC 3C 34 06 37 10 34 10 37 10 34 10 :1B
20D0 17 0F A8 32 66 39 EC 62 9E 08 58 49 EC 8B 83 00 :2E
20E0 00 10 2C 00 18 EC 62 58 49 9E 08 30 8B 36 10 EC :D6
20F0 62 9E 08 58 49 EC 8B C3 00 02 ED D1 39 EC 62 9E :C8
--------------------------------------------------------
Sum: 8F 20 C4 BF 2F C3 2B ED 27 0C 2A 92 76 C8 4D 78 :2E

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2100 0C 58 49 EC 8B 83 00 03 10 26 00 0F EC 62 58 49 :DE
2110 9E 0C 30 8B 36 10 CC 00 00 ED D1 EC 62 9E 0C 58 :85
2120 49 EC 8B 83 00 00 10 26 00 F2 DC 3C 83 CC 80 10 :62
2130 27 00 20 EC 62 58 49 9E 0A 30 8B 36 10 CC 00 02 :AD
2140 17 F5 46 36 06 CC 00 03 17 F3 54 C3 00 01 ED D1 :3D
2150 16 00 C9 DC 28 36 06 EC 62 9E 06 58 49 EC 8B A3 :CC
2160 C1 53 43 C3 00 01 DD 44 DC 2C 36 06 EC 62 9E 08 :74
2170 58 49 EC 8B A3 C1 53 43 C3 00 01 DD 46 DC 44 36 :4F
2180 06 CC 00 00 17 F3 A2 36 06 DC 46 36 06 CC 00 00 :E4
2190 17 F3 86 A4 C0 E4 C0 10 27 00 12 EC 62 58 49 9E :6E
21A0 0A 30 8B 36 10 CC 00 01 ED D1 16 00 6F DC 44 36 :71
21B0 06 CC 00 00 17 F3 62 36 06 DC 46 36 06 CC 00 00 :A4
21C0 17 F3 5E A4 C0 E4 C0 10 27 00 12 EC 62 58 49 9E :46
21D0 0A 30 8B 36 10 CC 00 02 ED D1 16 00 3F DC 44 36 :42
21E0 06 CC 00 00 17 F3 3A 36 06 DC 46 36 06 CC 00 00 :7C
21F0 17 F3 3E A4 C0 E4 C0 10 27 00 12 EC 62 58 49 9E :26
--------------------------------------------------------
Sum: CB 7E 9A 9E 99 CC D9 12 93 28 FD D1 42 E7 A1 AB :CF

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2200 0A 30 8B 36 10 CC 00 03 ED D1 16 00 0F EC 62 58 :63
2210 49 9E 0A 30 8B 36 10 CC 00 04 ED D1 EC 62 9E 06 :72
2220 58 49 EC 8B DD 48 EC 62 9E 08 58 49 EC 8B DD 4A :70
2230 EC 62 9E 08 58 49 EC 8B 83 00 04 10 2C 00 1A EC :D5
2240 62 58 49 9E 08 30 8B 36 10 EC 62 9E 08 58 49 EC :2B
2250 8B C3 00 02 ED D1 16 00 83 EC 62 58 49 9E 06 30 :6A
2260 8B 36 10 EC 62 9E 06 58 49 EC 8B 36 06 EC 62 9E :03
2270 0C 58 49 EC 8B 58 49 58 49 36 06 EC 62 9E 0A 58 :F0
2280 49 EC 8B E3 C1 9E 10 58 49 EC 8B E3 C1 ED D1 EC :78
2290 62 58 49 9E 08 30 8B 36 10 EC 62 9E 08 58 49 EC :2B
22A0 8B 36 06 EC 62 9E 0C 58 49 EC 8B 58 49 58 49 36 :4F
22B0 06 EC 62 9E 0A 58 49 EC 8B E3 C1 9E 12 58 49 EC :F5
22C0 8B E3 C1 ED D1 EC 62 58 49 9E 0C 30 8B 36 10 EC :73
22D0 62 9E 0C 58 49 EC 8B C3 00 01 ED D1 DC 4A 83 00 :4F
22E0 00 10 2F 00 11 DC 48 36 06 DC 4A 34 06 37 10 34 :8B
22F0 10 17 00 C4 32 64 EC 62 9E 08 58 49 EC 8B 83 00 :10
--------------------------------------------------------
Sum: 54 30 F9 85 44 66 E9 27 4D 01 88 37 49 90 84 C0 :E6

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2300 18 10 2C 00 28 EC 62 9E 06 58 49 EC 8B 36 06 EC :AE
2310 62 9E 08 58 49 EC 8B 36 06 DC 3C 34 06 37 10 34 :29
2320 10 37 10 34 10 17 0D 53 32 66 16 00 09 EC 62 34 :4B
2330 06 17 FC 1E 32 62 39 CC 00 00 9E 10 ED 84 CC 00 :BB
2340 02 9E 10 ED 02 CC 00 00 9E 10 ED 04 CC 00 00 9E :74
2350 10 ED 06 CC FF FE 9E 10 ED 08 CC 00 00 9E 10 ED :D6
2360 0A CC 00 02 9E 10 ED 0C CC FF FE 9E 10 ED 0E CC :BD
2370 00 00 9E 10 ED 88 10 CC 00 00 9E 12 ED 84 CC 00 :EC
2380 01 9E 12 ED 02 CC FF FE 9E 12 ED 04 CC FF FE 9E :71
2390 12 ED 06 CC 00 01 9E 12 ED 08 CC 00 02 9E 12 ED :E2
23A0 0A CC FF FF 9E 12 ED 0C CC FF FF 9E 12 ED 0E CC :BE
23B0 00 02 9E 12 ED 88 10 39 EC 62 36 06 EC 64 34 06 :84
23C0 37 10 34 10 17 00 42 32 64 EC 62 C3 00 01 36 06 :C8
23D0 EC 64 34 06 37 10 34 10 17 00 2E 32 64 EC 62 36 :74
23E0 06 EC 64 C3 00 01 34 06 37 10 34 10 17 00 1A 32 :42
23F0 64 EC 62 C3 00 01 36 06 EC 64 C3 00 01 34 06 37 :37
--------------------------------------------------------
Sum: 56 F8 D7 DB 1A 2C 48 7E 76 8C 03 91 98 FB 38 AD :1A

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2400 10 34 10 17 00 03 32 64 39 EC 64 83 00 18 10 2F :67
2410 00 01 39 EC 64 36 06 CC 00 28 17 F0 82 36 06 EC :6B
2420 62 E3 C1 9E 02 58 49 EC 8B DD 26 EC 62 36 06 EC :37
2430 64 34 06 37 10 34 10 17 09 FD 32 64 EC 64 36 06 :68
2440 CC 00 28 17 F0 59 36 06 EC 62 E3 C1 9E 00 58 49 :C1
2450 EC 8B 17 F2 09 39 EC 62 36 06 EC 64 36 06 CC 00 :A4
2460 28 17 F0 3B E3 C1 58 49 9E 00 30 8B 36 10 EC 66 :A0
2470 ED D1 EC 62 36 06 EC 64 36 06 CC 00 28 17 F0 1F :EE
2480 E3 C1 58 49 9E 02 30 8B 36 10 EC 68 ED D1 EC 68 :4C
2490 DD 26 EC 62 36 06 EC 64 34 06 37 10 34 10 17 09 :C2
24A0 96 32 64 EC 66 17 F1 B6 39 EC 64 83 00 01 36 06 :85
24B0 CC 00 03 17 F0 05 DD 40 EC 62 36 06 DC 40 58 49 :3F
24C0 E3 C1 83 00 11 47 56 47 56 DD 42 39 DC 28 36 06 :0A
24D0 DC 2C 36 06 CC 00 02 36 06 CC 00 02 36 06 EC 62 :A6
24E0 9E 06 58 49 EC 8B 36 06 EC 62 9E 08 58 49 EC 8B :04
24F0 36 06 CC 00 02 36 06 CC 00 02 34 06 37 10 34 10 :D9
--------------------------------------------------------
Sum: 58 D1 B3 7B 7D 4A 75 7C 9A CD 6F BD A0 BE 25 9E :C3
                                6
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2500 37 10 34 10 37 10 34 10 37 10 34 10 37 10 34 10 :2C
2510 37 10 34 10 37 10 34 10 17 0D 9E 32 E9 00 10 10 :13
2520 27 00 4E EC 62 9E 0E 58 49 EC 8B 83 00 01 10 26 :41
2530 00 3A EC 62 9E 06 58 49 EC 8B 36 06 EC 62 9E 08 :74
2540 58 49 EC 8B 34 06 37 10 34 10 17 FE 6B 32 64 EC :DF
2550 62 34 06 17 F9 FC 32 62 DC 18 C3 00 32 DD 18 CC :E6
2560 00 32 34 06 17 05 B0 32 62 16 00 05 CC 00 01 DD :91
2570 20 39 CC 00 01 34 06 17 07 15 32 62 CC 00 00 DD :D0
2580 16 16 00 36 CC 00 00 36 06 DC 16 36 06 CC 00 0A :6E
2590 34 06 37 10 34 10 37 10 34 10 17 06 A8 32 66 CC :79
25A0 00 01 DD 3A 16 00 07 DC 3A C3 00 01 DD 3A 83 03 :AC
25B0 E8 2F F4 DC 16 C3 00 01 DD 16 83 00 64 2F C5 CC :5B
25C0 00 00 34 06 17 06 C8 32 62 39 17 07 75 DD 38 DC :70
25D0 30 83 00 02 10 26 00 0B 17 00 FF 17 04 E8 CC 00 :DB
25E0 00 DD 30 DC 30 83 00 00 10 26 00 51 CC 00 00 DD :CC
25F0 4C DC 38 83 00 33 10 26 00 0A CC 00 01 DD 4C CC :18
--------------------------------------------------------
Sum: 1D CA 38 D9 36 B4 03 02 D6 15 31 DC 76 8B 6D EA :37

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2600 00 01 DD 2E DC 38 83 00 39 10 26 00 0A CC 00 02 :EA
2610 DD 4C CC 00 01 DD 2E DC 38 83 00 37 10 26 00 0A :0F
2620 CC 00 03 DD 4C CC 00 02 DD 2E DC 38 83 00 31 10 :A9
2630 26 00 0A CC 00 04 DD 4C CC 00 02 DD 2E DC 28 DD :E3
2640 48 DC 2C DD 4A DC 28 36 06 DC 30 58 49 58 49 D3 :D8
2650 4C 9E 10 58 49 EC 8B E3 C1 DD 28 DC 2C 36 06 DC :DB
2660 30 58 49 58 49 D3 4C 9E 12 58 49 EC 8B E3 C1 DD :DA
2670 2C DC 4C 83 00 00 10 27 00 07 DC 30 C3 00 01 DD :C2
2680 30 DC 28 36 06 DC 48 17 EE B7 36 06 DC 2C 36 06 :D0
2690 DC 4A 17 EE AC AA C0 EA C0 10 27 00 11 DC 48 36 :8D
26A0 06 DC 4A 34 06 37 10 34 10 17 FD 0C 32 64 DC 28 :AB
26B0 36 06 DC 2C 36 06 CC C5 00 36 06 DC 2E 36 06 CC :5F
26C0 00 C0 17 ED DA E3 C1 83 00 C0 34 06 37 10 34 10 :4A
26D0 37 10 34 10 17 09 A4 32 66 39 DC 28 36 06 DC 2C :68
26E0 34 06 37 10 34 10 17 FD C0 32 64 DC 42 36 06 DC :65
26F0 40 36 06 CC 00 07 17 ED A6 E3 C1 83 00 07 9E 04 :C9
--------------------------------------------------------
Sum: B2 0F 74 44 18 46 14 A1 7D FB 16 17 8A 34 7E AE :1B

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2700 58 49 EC 8B DD 3E DC 22 83 00 01 10 26 00 30 DC :F7
2710 3E 83 00 01 10 26 00 27 DC 18 D3 1E DD 18 DC 32 :07
2720 C3 00 01 DD 32 DC 40 36 06 DC 42 36 06 CC 00 05 :56
2730 34 06 37 10 34 10 37 10 34 10 17 F5 EC 32 66 DC :BC
2740 22 83 00 02 10 26 00 53 DC 3E 83 00 01 10 26 00 :04
2750 1A DC 40 36 06 DC 42 36 06 CC 00 03 34 06 37 10 :1C
2760 34 10 37 10 34 10 17 F5 C0 32 66 DC 3E 83 00 03 :D3
2770 10 26 00 27 DC 18 D3 1E DD 18 DC 32 C3 00 01 DD :E6
2780 32 DC 40 36 06 DC 42 36 06 CC 00 05 34 06 37 10 :36
2790 34 10 37 10 34 10 17 F5 90 32 66 DC 22 83 00 03 :87
27A0 10 26 00 5A DC 3E 83 00 01 10 26 00 27 DC 40 36 :DD
27B0 06 DC 42 36 06 CC 00 05 34 06 37 10 34 10 37 10 :3D
27C0 34 10 17 F5 64 32 66 DC 32 C3 00 01 DD 32 DC 18 :21
27D0 D3 1E DD 18 DC 3E 83 00 05 10 26 00 21 DC 40 36 :31
27E0 06 DC 42 36 06 CC 00 01 34 06 37 10 34 10 37 10 :39
27F0 34 10 17 F5 34 32 66 DC 32 83 00 01 DD 32 DC 22 :BB
--------------------------------------------------------
Sum: CA 6F A1 F6 0F DE AA 14 80 C8 12 6D EB 74 AD B8 :06

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2800 83 00 04 10 26 00 7D DC 3E 83 00 01 10 26 00 1A :28
2810 DC 40 36 06 DC 42 36 06 CC 00 03 34 06 37 10 34 :36
2820 10 37 10 34 10 17 F5 01 32 66 DC 3E 83 00 03 10 :F0
2830 26 00 27 DC 40 36 06 DC 42 36 06 CC 00 05 34 06 :0A
2840 37 10 34 10 37 10 34 10 17 F4 DE 32 66 DC 18 D3 :5E
2850 1E DD 18 DC 32 C3 00 01 DD 32 DC 3E 83 00 05 10 :A6
2860 26 00 21 DC 40 36 06 DC 42 36 06 CC 00 01 34 06 :00
2870 37 10 34 10 37 10 34 10 17 F4 AE 32 66 DC 32 83 :F8
2880 00 01 DD 32 DC 22 83 00 05 10 26 00 A0 DC 3E 83 :09
2890 00 01 10 26 00 1A DC 40 36 06 DC 42 36 06 CC 00 :CF
28A0 03 34 06 37 10 34 10 37 10 34 10 17 F4 7B 32 66 :71
28B0 DC 3E 83 00 03 10 26 00 27 DC 40 36 06 DC 42 36 :A9
28C0 06 CC 00 05 34 06 37 10 34 10 37 10 34 10 17 F4 :32
28D0 58 32 66 DC 18 D3 1E DD 18 DC 32 C3 00 01 DD 32 :AB
28E0 DC 3E 83 00 05 10 26 00 21 DC 40 36 06 DC 42 36 :A5
28F0 06 CC 00 04 34 06 37 10 34 10 37 10 34 10 17 F4 :31
--------------------------------------------------------
Sum: 66 F0 71 72 A6 17 63 30 DE 6D 85 55 26 51 95 3F :F9
```

OCTOPUS GARDENマシン語ダンプ・リスト

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2900  28 32 66 DC 32 83 00 01 DD 32 DC 3E 83 00 04 10 :12
2910  26 00 1A DC 40 36 06 DC 42 36 06 CC 00 01 34 06 :F9
2920  37 10 34 10 37 10 34 10 17 F3 FE 32 66 39 DC 28 :F3
2930  36 06 DC 2C 34 06 37 10 34 10 17 FB 6C 32 64 DC :F9
2940  40 36 06 CC 00 01 17 EB D8 36 06 DC 42 36 06 DC :95
2950  40 17 EB D5 AA C0 EA C0 10 27 00 05 CC 00 02 DD :12
2960  20 DC 42 36 06 CC 00 01 17 EB B6 36 06 DC 30 36 :7D
2970  06 CC 00 01 17 EB CA A4 C0 E4 C0 10 27 00 05 CC :AF
2980  00 02 DD 20 DC 2C 83 00 16 10 2F 00 05 CC 00 02 :B2
2990  DD 20 39 CC 00 01 34 06 17 02 F4 32 62 CC 00 01 :AB
29A0  DD 16 16 00 3D CC 00 00 36 06 CC 00 64 17 EC D9 :5A
29B0  C3 00 64 36 06 CC 00 0A 34 06 37 10 34 10 37 10 :45
29C0  34 10 17 02 80 32 66 CC 00 01 DD 3A 16 00 07 DC :52
29D0  3A C3 00 01 DD 3A 83 0F A0 2F F4 DC 16 C3 00 01 :20
29E0  DD 16 83 00 32 2F BE CC 00 00 34 06 17 02 A0 32 :86
29F0  62 39 17 01 81 DC 1E C3 00 01 DD 1E DC 1E 83 00 :6A
-----------------------------------------------------------
Sum:  8B 97 04 F2 D3 83 B8 C7 60 E6 7B DA AE 20 02 D0 :28

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2A00  05 10 2F 00 0C CC 00 01 DD 1E DC 22 C3 00 01 DD :B7
2A10  22 DC 1E DD 1C DC 22 83 00 05 10 2F 00 05 CC 00 :AB
2A20  05 DD 22 CC 00 01 DD 16 DC 1C 36 06 DC 16 16 00 :00
2A30  10 DC 16 34 06 17 F5 1A 32 62 DC 16 C3 00 01 DD :89
2A40  16 A3 C4 2F EC 33 42 17 F1 94 CC 00 00 DD 32 CC :50
2A50  00 00 DD 20 CC 00 01 36 06 CC 00 01 34 06 37 10 :54
2A60  34 10 17 F4 BD 32 64 DC 2A DD 28 DC 34 DD 2C CC :92
2A70  00 00 DD 30 39 CC 00 01 DD 16 DC 1C 36 06 DC 16 :2C
2A80  16 00 3C DC 16 9E 08 58 49 EC 8B 83 00 00 10 2F :C4
2A90  00 1D DC 16 9E 06 58 49 EC 8B 36 06 DC 16 9E 08 :9F
2AA0  58 49 EC 8B 34 06 37 10 34 10 17 F9 0B 32 64 DC :6A
2AB0  16 34 06 17 F4 9C 32 62 DC 16 C3 00 01 DD 16 A3 :D7
2AC0  C4 2F C0 33 42 39 CC 00 01 DD 16 16 00 43 CC 00 :46
2AD0  01 34 06 17 01 B0 32 62 CC 00 01 DD 3A 16 00 07 :98
2AE0  DC 3A C3 00 01 DD 3A 83 01 90 2F F4 CC 00 00 34 :28
2AF0  06 17 01 92 32 62 CC 00 01 DD 3A 16 00 07 DC 3A :5B
-----------------------------------------------------------
Sum:  B1 A6 AE C0 2E 5F 68 D6 FD DB E9 E5 EE 66 25 A3 :52

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2B00  C3 00 01 DD 3A 83 01 90 2F F4 DC 16 C3 00 01 DD :A5
2B10  16 83 00 0A 2F B8 39 CC 00 01 34 06 17 01 70 32 :84
2B20  62 CC 00 00 DD 4E 16 00 3D CC 00 00 36 06 DC 4E :DE
2B30  36 06 CC 00 0A 34 06 37 10 34 10 37 10 34 10 17 :79
2B40  01 03 32 66 CC 00 01 DD 50 EC 62 36 06 DC 50 16 :62
2B50  00 07 DC 50 C3 00 01 DD 50 A3 C4 2F F5 33 42 DC :00
2B60  4E C3 00 01 DD 4E 83 00 C8 2F BE CC 00 00 34 06 :7B
2B70  17 01 1C 32 62 39 CC 00 01 34 06 17 01 11 32 62 :C5
2B80  CC 01 2C DD 16 16 00 4F CC 00 05 36 06 CC 00 07 :31
2B90  17 EA F6 C3 00 01 34 06 37 10 34 10 17 02 53 32 :1E
2BA0  64 CC 00 00 36 06 DC 16 36 06 CC 00 0A 34 06 37 :E1
2BB0  10 34 10 37 10 34 10 17 00 8B 32 66 CC 00 01 DD :C3
2BC0  3A 16 00 07 DC 3A C3 00 01 DD 3A 83 0B B8 2F F4 :B1
2BD0  DC 16 C3 FF F6 DD 16 83 00 32 2C AC CC 00 00 34 :2A
2BE0  06 17 00 AB 32 62 CC 00 05 36 06 CC 00 05 34 06 :74
2BF0  37 10 34 10 17 01 FB 32 64 CC 00 01 DD 3A 16 00 :2E
-----------------------------------------------------------
Sum:  81 61 20 68 95 0F 67 84 88 99 AD 43 C3 54 28 49 :92

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2C00  1B CC 00 01 DD 16 16 00 07 DC 16 C3 00 01 DD 16 :A1
2C10  83 27 10 2F F4 DC 3A C3 00 01 DD 3A 83 00 05 2F :85
2C20  E0 39 EC 62 F7 FD 0E CC 00 03 F7 FD 0D CC 00 00 :05
2C30  F7 FD 0D EC 64 F7 FD 0E CC 00 02 F7 FD 0D CC 00 :EE
2C40  00 F7 FD 0D 39 EC 62 58 49 36 06 EC 64 34 06 37 :26
2C50  10 34 10 17 FF CC 32 64 EC 62 58 49 C3 00 01 36 :B5
2C60  06 EC 64 1F 89 4F 34 06 37 10 34 10 17 FF B3 32 :0D
2C70  64 EC 62 C3 00 08 36 06 EC 66 34 06 37 10 34 10 :D0
2C80  17 FF 9F 32 64 39 EC 62 C3 00 80 F7 FD 03 39 CC :11
2C90  00 07 36 06 CC 00 FF 36 06 CC 00 FF 36 06 EC 62 :9F
2CA0  A3 C1 53 43 C3 00 01 A4 C0 E4 C0 34 06 37 10 34 :7B
2CB0  10 17 FF 6E 32 64 39 17 01 66 CC FC 82 DD 52 9E :F8
2CC0  52 CC 00 3F E7 80 CC 00 59 E7 80 CC 00 41 E7 80 :C4
2CD0  CC 00 4D E7 80 CC 00 41 E7 80 CC 00 55 E7 80 CC :48
2CE0  00 43 E7 80 CC 00 48 E7 80 CC 00 49 E7 80 CC 00 :6D
2CF0  93 E7 80 CC D3 8F ED 81 CC 00 90 E7 80 9F 52 9E :E8
-----------------------------------------------------------
Sum:  6A 00 B7 DF 18 6D 7F 61 41 37 9A 5E 79 81 A8 DE :55

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2D00  52 CC 00 B7 E7 80 CC D4 0A ED 81 CC 00 B6 E7 80 :3D
2D10  CC D4 09 ED 81 CC 00 86 E7 80 EC 64 E7 80 CC 00 :53
2D20  B7 E7 80 EC 62 ED 81 CC 00 B6 E7 80 CC D4 0A ED :5A
2D30  81 CC 00 39 E7 80 9F 52 17 00 F5 17 00 E2 17 00 :FA
2D40  EF 39 17 00 DB CC FC 82 DD 52 9E 52 CC 00 29 E7 :5F
2D50  80 CC 00 00 E7 80 9F 52 17 00 D5 17 00 C2 4F F6 :AE
2D60  FC 83 DD 54 CC 00 80 F7 FC 80 17 00 C3 DC 54 39 :B2
2D70  CC FC 82 DD 52 17 00 A8 9E 52 CC 00 01 E7 80 EC :48
2D80  62 E7 80 EC 64 E7 80 EC 66 E7 80 EC 68 E7 80 EC :E0
2D90  6A E7 80 EC 6C E7 80 EC 6E E7 80 EC E8 10 E7 80 :9C
2DA0  9F 52 17 00 8B 39 CC 00 00 36 06 CC 00 28 36 06 :04
2DB0  CC 00 19 36 06 CC 00 00 36 06 CC 00 19 36 06 CC :16
2DC0  00 00 36 06 CC 00 01 36 06 CC 00 00 34 06 37 10 :92
2DD0  34 10 37 10 34 10 37 10 34 10 37 10 34 10 37 10 :2C
2DE0  34 10 37 10 34 10 37 10 34 10 17 FF 83 32 E9 00 :0E
2DF0  10 39 CC FD 38 36 06 EC 62 E3 C1 36 06 EC 64 E7 :EB
-----------------------------------------------------------
Sum:  3C 50 9F 2B 5E 45 48 05 70 20 80 19 9D FA 7E B4 :38

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2E00  D1 39 17 00 1B CC FC 82 DD 52 9E 52 CC 00 02 E7 :5A
2E10  80 EC 62 E7 80 CC 00 00 E7 80 9F 52 17 00 11 39 :BA
2E20  B6 FD 05 2B FB 86 80 B7 FD 05 B6 FD 05 2A FB 39 :B3
2E30  CC 00 00 F7 FD 05 39 17 FF E6 CC FC 82 DD 52 9E :11
2E40  52 CC 00 03 E7 80 CC 00 05 E7 80 CC 00 12 E7 80 :05
2E50  EC 62 E7 80 EC 64 E7 80 CC 00 11 E7 80 CC 00 80 :FC
2E60  D3 26 E7 80 9F 52 17 FF C7 39 CC 00 00 DD 56 16 :7C
2E70  00 26 CC C4 00 D3 56 36 06 30 8D 00 BA 1F 10 D3 :94
2E80  56 1F 01 4F E6 84 34 06 37 10 34 10 17 FE 28 32 :63
2E90  64 DC 56 C3 00 01 DD 56 83 00 45 2F D5 CC 00 00 :25
2EA0  DD 56 16 00 26 CC C4 50 D3 56 36 06 30 8D 02 A0 :13
2EB0  1F 10 D3 56 1F 01 4F E6 84 34 06 37 10 34 10 17 :0D
2EC0  FD F5 32 64 DC 56 C3 00 01 DD 56 83 00 45 2F D5 :7D
2ED0  CC 00 00 DD 56 16 00 26 CC C4 A0 D3 56 36 06 30 :00
2EE0  8D 02 B0 1F 10 D3 56 1F 01 4F E6 84 34 06 37 10 :F1
2EF0  34 10 17 FD C2 32 64 DC 56 C3 00 01 DD 56 83 00 :5C
-----------------------------------------------------------
Sum:  24 04 51 95 34 EF 76 B8 93 5A 3A A7 37 43 D6 DE :5B

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
2F00  0F 2F D5 CC 00 00 DD 56 16 00 26 CC C4 B0 D3 56 :B7
2F10  36 06 30 8D 02 8C 1F 10 D3 56 1F 01 4F E6 84 34 :EC
2F20  06 37 10 34 10 17 FD 8F 32 64 DC 56 C3 00 01 DD :9D
2F30  56 83 00 28 2F D5 39 BE D3 8F FE D3 91 B6 D4 09 :53
2F40  8D 0A 33 C9 40 00 8D 04 33 C9 40 00 EC 81 ED C4 :BE
2F50  EC 81 ED C8 50 EC 81 ED C9 00 A0 EC 81 ED C9 00 :58
2F60  F0 EC 81 ED C9 01 40 EC 81 ED C9 01 90 EC 81 ED :62
2F70  C9 01 E0 EC 81 ED C9 02 30 39 CC 00 00 DD 56 16 :4D
2F80  00 21 EC 64 D3 56 36 06 EC 62 D3 56 1F 01 4F E6 :A2
2F90  84 34 06 37 10 34 10 17 FD 1D 32 64 DC 56 C3 00 :05
2FA0  01 DD 56 83 00 2F 2F DA 39 CC 00 00 DD 56 16 00 :3D
2FB0  21 EC 64 D3 56 36 06 EC 62 D3 56 1F 01 4F E6 84 :26
2FC0  34 06 37 10 34 10 17 FC EE 32 64 DC 56 C3 00 01 :52
2FD0  DD 56 83 00 BF 2F DA 39 EC 62 36 06 CC 00 00 17 :24
2FE0  E5 3F 36 06 EC 62 36 06 CC 00 27 17 E5 3B AA C0 :7E
2FF0  EA C0 36 06 EC 64 36 06 CC 00 00 17 E5 23 AA C0 :C7
-----------------------------------------------------------
Sum:  59 E0 68 2C 1F 46 21 B6 91 EA B0 CC 29 A0 1B 39 :1D

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3000  EA C0 36 06 EC 64 36 06 CC 00 18 17 E5 1B AA C0 :D7
3010  EA C0 10 27 00 01 39 17 FE 06 CC FC 82 DD 52 9E :4D
3020  52 CC 00 3F E7 80 CC 00 59 E7 80 CC 00 41 E7 80 :C4
3030  CC 00 4D E7 80 CC 00 41 E7 80 CC 00 55 E7 80 CC :48
3040  00 43 E7 80 CC 00 48 E7 80 CC 00 49 E7 80 CC 00 :6D
3050  93 E7 80 CC C4 00 ED 81 CC 00 90 E7 80 EC 66 ED :FA
3060  81 EC 64 36 06 CC 02 80 17 E4 34 36 06 EC 62 58 :6C
3070  49 E3 C1 ED 81 9F 52 17 FD B6 39 EC 62 36 06 CC :A5
3080  00 00 17 E4 9C 36 06 EC 62 36 06 CC 00 26 17 E4 :4A
3090  98 AA C0 EA C0 36 06 EC 64 36 06 CC 00 00 17 E4 :3B
30A0  80 AA C0 EA C0 36 06 EC 64 36 06 CC 00 17 17 E4 :3A
30B0  78 AA C0 EA C0 10 27 00 01 39 17 FD 63 CC FC 82 :BE
30C0  DD 52 EC 64 36 06 CC 02 80 17 E3 D3 36 06 EC 62 :60
30D0  58 49 E3 C1 ED 62 9E 52 CC 00 3F E7 80 CC 00 59 :1B
30E0  E7 80 CC 00 41 E7 80 CC 00 4D E7 80 CC 00 41 E7 :4F
30F0  80 CC 00 55 E7 80 CC 00 43 E7 80 CC 00 48 E7 80 :F9
-----------------------------------------------------------
Sum:  7B 2A 11 DE 91 9D B3 41 24 F9 DF 98 70 D1 52 0B :E8

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3100  CC 00 49 E7 80 CC 00 93 E7 80 CC C4 B0 ED 81 CC :BC
3110  00 90 E7 80 9F 52 9E 52 EC 66 ED 81 EC 62 ED 81 :54
3120  EC 66 C3 00 30 ED 81 EC 62 C3 00 02 ED 81 EC 66 :86
3130  C3 00 60 ED 81 EC 62 C3 02 80 ED 81 EC 66 C3 00 :A7
3140  90 ED 81 EC 62 C3 02 82 ED 81 9F 52 17 FC E1 39 :1F
3150  B6 D4 09 BE D3 8F FE D3 91 8D 0A 30 89 40 00 8D :32
3160  04 30 89 40 00 EC 84 ED C1 EC 88 50 ED C1 EC 89 :02
3170  00 A0 ED C1 EC 89 00 F0 ED C1 EC 89 01 40 ED C1 :C5
3180  EC 89 01 90 ED C1 EC 89 01 E0 ED C1 EC 89 02 30 :5F
3190  ED C1 39 4F BE D3 8F C6 30 AB 80 5A 26 FB B7 D3 :7C
31A0  90 39 BE D3 8F FE D3 91 BD C4 06 BE D3 93 FE D3 :C7
31B0  95 BD C4 06 BE D3 97 FE D3 99 BD C4 06 BE D3 9B :61
31C0  FE D3 9D BD C4 06 39 17 FC 56 CC FC 82 DD 52 9E :AE
31D0  52 CC 00 15 E7 80 DC 26 E7 80 CC 00 00 E7 80 DC :12
31E0  58 ED 81 DC 5A ED 81 EC 62 ED 81 EC 64 ED 81 EC :D0
31F0  66 E7 80 9F 52 17 FC 38 EC 62 DD 58 EC 64 DD 5A :13
-----------------------------------------------------------
Sum:  D1 3A AD 04 40 AD 7C 05 55 F1 E9 00 C0 5D 91 F4 :FB

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3200  39 17 FC 1C CC FC 82 DD 52 9E 52 CC 00 19 E7 80 :1D
3210  DC 26 E7 80 CC 00 00 E7 80 CC 00 00 E7 80 EC 66 :21
3220  E7 80 EC 68 E7 80 EC 62 ED 81 EC 64 ED 81 CC 00 :68
3230  06 E7 80 EC 6A E7 80 EC 6C E7 80 EC 6E E7 80 EC :96
3240  E8 10 E7 80 EC E8 12 E7 80 EC E8 14 E7 80 9F 52 :EC
3250  17 FB DD 39 EC 62 36 06 EC 64 E3 C1 36 06 EC 66 :34
3260  17 E2 C6 10 27 00 07 CC 00 01 39 16 00 04 CC 00 :E9
3270  00 39 EC 62 36 06 EC 64 36 06 EC 66 34 06 37 10 :22
3280  34 10 37 10 34 10 17 FF CB 32 66 36 06 EC 66 36 :0C
3290  06 EC 68 36 06 EC 62 34 06 37 10 34 10 37 10 34 :24
32A0  10 17 FF B0 32 66 A4 C0 E4 C0 10 27 00 07 CC 00 :80
32B0  01 39 16 00 04 CC 00 00 39 EC 62 36 06 EC 66 36 :6B
32C0  06 EC 6A 36 06 EC 6E 34 06 37 10 34 10 37 10 34 :32
32D0  10 37 10 34 10 17 FF 9A 32 68 36 06 EC 64 36 06 :AD
32E0  EC 68 36 06 EC 6C 36 06 EC E8 10 34 06 37 10 34 :BD
32F0  10 37 10 34 10 37 10 34 10 17 FF 76 32 68 A4 C0 :B0
-----------------------------------------------------------
Sum:  75 D8 39 B5 A0 87 F9 2A EF DC EB 18 E3 E1 4F 68 :CE

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3300  E4 C0 10 27 00 07 CC 00 01 39 16 00 04 CC 00 00 :CE
3310  39 80 00 80 00 80 00 80 00 80 00 C0 00 C0 00 C0 :F9
3320  00 F0 00 FE 00 FF C0 FF F8 FF FF C0 00 C0 00 C0 :E2
3330  00 80 00 80 00 80 00 80 00 80 00 C0 00 C0 00 C0 :C0
3340  00 00 00 00 00 00 3F 01 F0 07 80 0F 00 0F 00 00 :D5
3350  00 00 00 00 00 00 3F 01 F0 07 80 0F 00 0F 00 00 :D5
3360  00 00 1F 03 FF 1F FF 3F FF 7F FF FF FF FF FF FF :F6
3370  FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :FF
3380  01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :01
3390  01 F8 00 FF C0 FF F8 FF FC FF FE FF FF FF FF FF :A2
```

OCTOPUS GARDENマシン語ダンプ・リスト

```
33A0 FE 0F 00 0F 00 07 80 00 00 00 00 00 00 00 00 00 :A3
33B0 01 0F 00 0F 00 07 80 00 00 00 00 00 00 00 00 00 :A6
33C0 01 FF FF FF FF FF FF 7F FF 3F FF 1F FF 03 FF 00 :D7
33D0 1E 00 01 00 03 00 07 00 0E 00 1C 00 78 0F C0 F8 :92
33E0 00 00 01 00 03 00 07 00 0E 00 1C 00 78 0F C0 F8 :74
33F0 00 FF FE FF FC FF F8 FF F0 FF E0 FF 80 F0 00 00 :2C
--------------------------------------------------------
Sum: 3C C4 2E 43 C0 30 06 BD DF 02 29 7A 71 39 7D 2E :FD

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3400 00 00 00 00 00 00 00 00 F8 00 3C 00 3C 78 F8 88 :68
3410 00 00 1F 00 FF 03 FF 07 FF 0E 3F 0E 3F 7F FF 89 :C7
3420 FE 00 00 00 00 00 00 00 F8 00 3C 00 3C 78 F8 8E :6C
3430 01 00 00 00 00 00 00 3E 18 0F 0C 0F 0C 3E 00 00 :CB
3440 0C FF 00 FF E0 FF FC FF FE 8F FF 8F FF FF FF FF :FB
3450 FF 00 00 00 00 00 00 3E 18 0F 0C 0F 0C 3E 00 00 :C9
3460 0C 88 00 88 00 70 00 00 00 00 00 00 00 00 00 00 :8C
3470 00 8E 01 8F FF 70 00 00 00 7F 70 66 E0 E6 FF CE 7F :F4
3480 1C 89 FE 88 00 7F FF 00 00 00 00 00 00 00 00 00 :A9
3490 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
34A0 00 FF FF BF FE 7F F8 FF C0 39 86 71 C7 71 FF 38 :90
34B0 7C 00 00 40 00 80 00 00 00 00 00 00 00 00 00 00 :3C
34C0 00 00 00 00 00 00 00 18 7C 30 F0 30 F0 00 7C 30 :80
34D0 00 00 FF 07 FF 3F FF 7F FF FF F1 FF F1 FF FF FF :9E
34E0 FF 00 00 00 00 00 00 00 18 7C 30 F0 30 F0 00 7C 30 :7F
34F0 00 00 00 00 00 00 00 1F 00 3C 00 3C 00 1F 1E 00 :D4
--------------------------------------------------------
Sum: AD 9D 1C A4 DB 9F F1 4F 53 FF 8B A7 4C 78 D0 B4 :90

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3500 11 F8 00 FF 00 FF C0 FF E0 FC 70 FC 70 FF FE 7F :FA
3510 91 00 00 00 00 00 00 00 1F 00 3C 00 3C 00 1F 1E 80 :E5
3520 71 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :71
3530 00 FF FF 7F FD 1F FE 03 FF 61 9C E3 8E FF 8E 3E :D2
3540 3C 00 00 00 02 00 01 00 00 00 00 00 00 00 00 00 :3F
3550 00 00 11 00 11 00 0E 00 00 00 00 00 00 00 00 00 :30
3560 00 80 71 FF F1 00 0E FE 00 66 0E 67 07 73 FF 38 :79
3570 FE 7F 91 00 11 FF FE 00 00 00 00 00 00 00 00 00 :1C
3580 00 00 00 00 00 00 00 18 00 30 00 30 00 00 00 30 :A8
3590 00 00 FF 07 FF 3E 1F 7F E1 FF FC FF E1 FE 1F FF :B9
35A0 FF 00 00 00 00 00 00 18 00 30 00 30 00 00 00 30 :A7
35B0 00 00 00 00 00 00 00 00 00 00 00 00 00 1E 00 :1E
35C0 11 F8 00 FF 00 F9 C0 87 E0 3F F0 87 F0 F8 FE 7F :43
35D0 91 00 00 00 00 00 00 00 00 00 00 00 00 00 1E 80 :2F
35E0 71 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :71
35F0 00 FF FF 7F FD 1F FE 03 FF 61 9C E3 8E FF 8E 3E :D2
--------------------------------------------------------
Sum: 5F ED 10 02 0E 73 B6 58 9F FE A2 4B 64 85 90 11 :01

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3600 3C 00 00 00 02 00 01 00 00 00 00 00 00 00 00 00 :3F
3610 00 00 11 00 11 00 0E 00 00 00 00 00 00 00 00 00 :30
3620 00 80 71 FF F1 00 0E FE 00 66 0E 67 07 73 FF 38 :79
3630 FE 7F 91 00 11 FF FE 00 00 00 00 00 00 00 00 00 :1C
3640 00 FF FF FF FF FF FF FF FF FF FF FF F3 FF E1 FF :C7
3650 F3 00 01 00 07 00 1F 00 7F 03 FF 1F F3 FF E1 00 :8D
3660 73 00 01 00 07 00 1F 00 7F 03 FF 1F F3 FF E1 00 :0D
3670 73 FF FF FF FF FF FF FF FF FF FF CF FF 87 FF CF :8C
3680 FF 80 00 E0 00 F8 00 FE 00 FF C0 CF F8 87 FF CE :2F
3690 00 80 00 E0 00 F8 00 FE 00 FF C0 CF F8 87 FF CE :30
36A0 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F1
36B0 FF 00 7F 00 7F 00 7F 00 7E 00 66 00 66 60 66 3F :CB
36C0 C6 00 7F 00 7F 00 7F 00 7E 00 66 00 66 60 66 3F :92
36D0 C6 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :B7
36E0 FF FE 00 FE 00 FE 00 7E 30 66 18 66 18 66 18 63 :84
36F0 F0 FE 00 FE 00 FE 00 7E 30 66 18 66 18 66 18 63 :75
--------------------------------------------------------
Sum: 8C F7 0F B7 1D E7 53 F2 56 32 84 DB C9 8F 99 E4 :4E

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3700 F0 00 00 01 FF 00 7F 00 1F 00 07 00 1F 00 77 01 :2C
3710 C7 00 00 01 FF 00 7F 00 1F 00 07 00 1F 00 7F 01 :0B
3720 FF 00 00 01 FF 00 7F 00 1F 00 07 00 1F 00 7F 01 :43
3730 CF 00 00 FF 80 FE 00 F8 00 E0 00 F8 00 FA 00 30 :46
3740 80 00 00 FF 80 FE 00 F8 00 E0 00 F8 00 FE 00 FF :CA
3750 80 00 00 FF 80 FE 00 F8 00 E0 00 F8 00 FE 00 F3 :BE
3760 80 07 80 1E 10 78 60 E1 C2 67 86 3F 0E 0F 1E 00 :17
3770 FF 07 FF 1F FF 7F FF FF FF 7F FF 3F FF 0F FF 00 :69
3780 FF 07 92 1E 72 79 E6 E7 CE 7F 9E 3F 3E 0F FE 00 :E3
3790 FF 00 60 02 18 01 86 10 E1 18 7A 1C 3C 1F 30 3F :69
37A0 00 FF E0 FF F8 FF FE FF FF FF FF FF FC FF F0 FF :B7
37B0 00 49 E0 4E 78 67 9E 73 E7 79 FE 7C FC 7F F0 FF :AB
37C0 00 CC 00 00 34 06 17 F6 39 32 62 CC 00 01 DD 26 :B0
37D0 CC 00 00 DD 58 CC 00 90 DD 5A CC 02 7F 36 06 CC :E9
37E0 00 C7 36 06 CC 00 02 34 06 37 10 34 10 37 10 34 :11
37F0 10 17 F9 D3 32 66 CC 00 07 DD 26 CC 00 08 36 06 :71
--------------------------------------------------------
Sum: DE 07 60 60 10 09 C9 EB D6 35 12 0A 6B 36 C9 8E :91

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3800 CC 00 15 34 06 37 10 34 10 17 F6 2B 32 64 17 DC :67
3810 7A 20 4F 70 65 6E 2D 52 65 61 63 68 20 20 50 72 :3E
3820 65 73 65 6E 74 73 20 00 CC 00 01 DD 16 16 00 75 :FD
3830 DC 16 83 00 01 36 06 CC 00 10 36 06 CC CF 00 34 :99
3840 06 37 10 34 10 37 10 34 10 17 F8 2F 32 66 DC 16 :E4
3850 36 06 CC 00 10 36 06 CC C5 00 34 06 37 10 34 10 :AA
3860 37 10 34 10 17 F8 14 32 66 CC 00 01 DD 3A 16 00 :40
3870 14 DC 3A 84 00 C4 01 34 06 17 F4 0A 32 62 DC 3A :6C
3880 C3 00 01 DD 3A 83 03 E8 2F E7 CC 00 01 DD 3A 16 :59
3890 00 07 DC 3A C3 00 01 DD 3A 83 3A 98 2F F4 DC 16 :62
38A0 C3 00 01 DD 16 83 00 13 2F 86 CC 00 01 34 06 17 :20
38B0 F3 DD 32 62 CC 00 0A DD 3A 16 00 99 DC 3A 36 06 :52
38C0 CC 00 0A 17 DB F5 36 06 CC 00 0A 17 DB D1 93 3A :5F
38D0 10 26 00 49 CC 00 13 36 06 DC 3A 36 06 CC 00 0A :C2
38E0 17 DB D8 83 00 01 36 06 CC CF 00 34 06 37 10 34 :DA
38F0 10 37 10 34 10 17 F7 83 32 66 CC 00 13 36 06 DC :BB
```

```
--------------------------------------------------------
Sum: 8A EE 98 47 AD 8A 12 32 24 99 92 68 B3 C4 64 F4 :58

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3900 3A 36 06 CC 00 0A 17 DB B2 36 06 CC C6 80 34 06 :78
3910 37 10 34 10 37 10 34 10 17 F7 60 32 66 CC 00 00 :E8
3920 36 06 DC 3A 58 49 36 06 CC 00 0A 34 06 37 10 34 :BA
3930 10 37 10 34 10 17 F3 0D 32 66 CC 00 01 DD 16 16 :20
3940 00 07 DC 16 C3 00 01 DD 16 83 00 64 2F F4 DC 3A :D0
3950 C3 00 01 DD 3A 83 00 8C 10 2F FF 60 CC 00 00 34 :88
3960 06 17 F3 2B 32 62 CC 00 13 36 06 CC 00 10 36 06 :02
3970 CC CE 40 34 06 37 10 34 10 37 10 34 10 17 F6 FB :32
3980 32 66 17 F1 41 CC 00 05 DD 26 CC 00 50 36 06 CC :D9
3990 00 08 36 06 CC 00 0A 36 06 CC 00 05 36 06 CC 00 :2F
39A0 4F 36 06 CC 00 43 36 06 CC 00 54 36 06 CC 00 50 :4E
39B0 36 06 CC 00 55 36 06 CC 00 53 34 06 37 10 34 10 :7D
39C0 37 10 34 10 37 10 34 10 37 10 34 10 37 10 34 10 :2C
39D0 37 10 34 10 37 10 34 10 37 10 34 10 37 10 34 10 :2C
39E0 17 F8 1E 32 E9 00 14 CC 00 05 DD 26 CC 00 50 36 :82
39F0 06 CC 00 38 36 06 CC 00 0A 36 06 CC 00 05 36 06 :65
--------------------------------------------------------
Sum: 8E FD DB E9 C3 01 DF 94 37 52 F0 49 3B B8 56 47 :D8

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3A00 CC 00 47 36 06 CC 00 41 36 06 CC 00 52 36 06 CC :BE
3A10 00 44 36 06 CC 00 45 36 06 CC 00 4E 34 06 37 10 :68
3A20 34 10 37 10 34 10 37 10 34 10 37 10 34 10 37 10 :2C
3A30 34 10 37 10 34 10 37 10 34 10 37 10 34 10 37 10 :2C
3A40 34 10 17 F7 BC 32 E9 00 14 CC 00 07 DD 26 CC 00 :DF
3A50 19 36 06 CC 00 0F 34 06 37 10 34 10 17 F3 D8 32 :09
3A60 64 17 DA 27 49 6E 73 74 72 75 63 74 69 6F 6E 20 :3E
3A70 3F 00 17 DA E1 DD 3E DC 3E 36 06 CC 00 6E 17 DA :AD
3A80 B8 36 06 DC 3E 36 06 CC 00 4E 17 DA AC AA C0 EA :55
3A90 C0 10 27 00 01 39 CC 00 00 34 06 17 F3 64 32 62 :39
3AA0 CC 00 07 DD 26 CC 00 10 36 06 CC 00 00 34 06 37 :2B
3AB0 10 34 10 17 F3 81 32 64 17 D9 D0 4D 49 53 53 49 :BA
3AC0 4F 4E 3A 00 CC 00 05 36 06 CC 00 02 34 06 37 10 :33
3AD0 34 10 17 F3 62 32 64 17 D9 B1 43 68 61 6E 67 65 :2D
3AE0 20 54 68 65 20 43 6F 6C 6F 72 20 4F 66 20 54 68 :11
3AF0 65 20 47 61 72 64 65 6E 00 CC 00 16 36 06 CC 00 :C0
--------------------------------------------------------
Sum: 80 0D 3D A9 38 0D C2 54 3A 95 F3 D2 64 81 DD D1 :F5

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3B00 03 34 06 37 10 34 10 17 F3 2D 32 64 17 D9 7C 54 :55
3B10 6F 20 4C 69 67 68 74 20 42 6C 75 65 00 CC 00 05 :00
3B20 36 06 CC 00 05 34 06 37 10 34 10 17 F3 09 32 64 :7B
3B30 17 D9 58 4B 65 79 00 CC 00 05 36 06 CC 00 06 34 :84
3B40 06 37 10 34 10 17 F2 EF 32 64 17 D9 3E 46 75 6E :76
3B50 63 74 69 6F 6E 3A 00 CC 00 10 36 06 CC 00 06 34 :75
3B60 06 37 10 34 10 17 F2 CF 32 64 17 D9 1E 37 20 20 :84
3B70 20 20 39 00 CC 00 11 36 06 CC 00 07 34 06 37 10 :E6
3B80 34 10 17 F2 B2 32 64 CC 00 EF 17 DA D1 17 DB FB :FC
3B90 20 20 00 CC 00 EE 17 DA C5 CC 00 12 36 06 CC 00 :96
3BA0 08 36 06 CC C5 00 34 06 37 10 34 10 37 10 34 10 :25
3BB0 17 F4 C8 32 66 CC 00 11 36 06 CC 00 0A 34 06 37 :CB
3BC0 10 34 10 17 F2 71 32 64 CC 00 EE 17 DA 90 17 D8 :8E
3BD0 BA 20 20 00 CC 00 EF 17 DA 84 CC 00 10 36 06 CC :0E
3BE0 00 0B 34 06 37 10 34 10 17 F2 4C 32 64 17 DB 9B :45
3BF0 31 20 20 20 20 33 00 CC 00 0B 36 06 CC 00 0E 34 :05
--------------------------------------------------------
Sum: BC 0E A1 BB 2D 51 83 0E 9E C8 A4 F0 94 6F 67 78 :11

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
3C00 06 37 10 34 10 17 F2 2F 32 64 17 D8 7E 42 4F 4E :AB
3C10 55 53 3A 00 CC 00 12 36 06 CC 00 0E 36 06 CC CB :A9
3C20 C0 34 06 37 10 34 10 37 10 34 10 17 F4 4D 32 66 :00
3C30 CC 00 15 36 06 CC 00 0E 34 06 37 10 34 10 17 F1 :C4
3C40 F6 32 64 17 DB 45 35 30 30 20 50 4F 49 4E 54 53 :52
3C50 00 CC 00 05 36 06 CC 00 11 34 06 37 10 34 10 17 :C6
3C60 F1 D5 32 64 17 D8 24 44 6F 6E 27 74 20 74 6F 75 :A3
3C70 63 68 00 CC 00 0A 36 06 CC 00 13 36 06 C6 80 :0A
3C80 34 06 37 10 34 10 37 10 34 10 17 F3 EE 32 66 CC :AC
3C90 00 18 36 06 CC 00 11 34 06 37 10 34 10 17 F1 97 :95
3CA0 32 64 17 D7 E6 44 6F 6E 27 74 20 72 69 63 6B 00 :EF
3CB0 CC 00 1C 36 06 CC 00 13 36 06 CC CC 80 34 06 37 :C8
3CC0 10 34 10 37 10 34 10 17 F3 B1 32 66 CC 00 0F 36 :43
3CD0 06 CC 00 17 34 06 37 10 34 10 17 F1 5A 32 64 17 :BD
3CE0 D7 A9 48 69 74 20 41 6E 79 20 4B 65 79 00 17 D8 :25
3CF0 65 DD 3E 39 10 DE FE 35 FF 00 00 00 00 00 00 00 :D9
--------------------------------------------------------
Sum: B5 01 31 00 CB 9C AC B3 2E CE 95 5E E1 79 4F 8E :D3
```

バトミントン・ゲーム

サブにDATAを送って高速PUT

■古田　昌

このゲームは自己流の高速表示システムをKコンパイラ化して作成した，オリジナルな高速ゲームです．初めは「テニス・ゲーム」を作るつもりだったのが，なぜか『バドミントン・ゲーム』になってしまいました．

遊び方

35, 36行めのFSおよびMSが，"C" にセットされていると，左および右のプレーヤーをコンピュータが自動的に操作します．この場合，370行めおよび490行めの**RND(10)**によってランダムにミスをするように作成してあります．

人が操作する場合は左側が1，2，3キーが右側7，8，9キーで操作します．それぞれ左移動，停止およびショット，右移動となります．なお，羽根が向かってくるときだけ操作が可能で，羽根を打つたびにキーの入力情報はクリアされます．

羽根は放物線を描いて飛んでくるので，うまく羽根の下に移動して打ち返してください．サービスは4，6キーで，フォルトはとりません．ジュースなしで１方が15点先取すると，ゲームは終了します．再ゲームは１から９までの適当なキーを押してください．プレイは必ず左から初まります．

FGETとFPUTについて

まず，40050からのデータを**FGET**〔**ADDRES**，**0**〕でサブに転送します．つぎに表示したいデータを**FGET**〔**ADDRES, No.**〕によって転送します．これで**ADDRES**から256バイトのデータがサブメモリに書き込まれます．

これを表示するのは**FPUT**〔X座標，Y座標，横長，縦長No.，**TYP**〕を使います．ここでX座標および横長はバイト座標を使っているので，０～639ではなく０～79の値です．Y座標と縦長はドットと同じです．

No.では**FGET**のときに定めたNo.を使います．TYPについては１～７がそのカラーコードで表示，８以上はカラーデータ込みとして表示します．この場合のDATA形式はF-BASICの場合と同一です．ただし，いまのところ**PSET**しか使えません．

写真１　彼女，ミスして！

行番号マップ

行番号	内　容
10～100	初期設定
120～160	点表示
170～	ゲーム・セット判断
180～200	ゲーム再開待
250～270	サービス待
300～320	シャトルの移動
330～	左右判断
340～430	右プレイ
450～550	左プレイ
570～630	ショット判断
670～	シャトルの表示，300へ戻る
700～780	ミス・ショット，120へ戻る
800～830	シャトル表示のサブルーチン
840～	ラケット消去ルーチン
850～	軌道の再定義
900～950	サブメモリへのデータ転送
10000～100090	サブシステム
40000～40220	高速表示システム
68FF H	INKEY バッファ
6900H～742FH	データ

キャラクタについて

●人物については一切のモデルはありません．
●キャラクタの作成は方眼紙に絵を書いて作りました（１週間以上もかかった!）．
●マイコンでこれだけのキャラクタをこれだけのスピードで動かせるのですから，技術の進歩は恐ろしいような気がします．

最後に

このゲームは僕のオリジナルです．皆さんも自分なりのゲームを作ってみてはいかがですか？

◘参考文献
1)FM-8活用研究
2)テレビ・テニス

リスト1 バドミントン・ゲーム Kコンパイラ

```
10 '(* <<<<<< ショキ セッテイ >>>>>>>> *)
20 'PRINT / "COM vs COM ...0"
21 'PRINT / "COM vs MAN ...1"
22 'PRINT / "MAN vs MAN....2"
23 'PRINT / "SELECT NUMBER "
24 'FS=0;MS=0
30 'SIN;NN=INPUT
34 'IF NN>2 OR NN<0 THEN GOTO SIN;FI
35 'IF NN<=1 THEN FS='C';FI
36 'IF NN=0 THEN MS='C';FI
40 'CLS[0];GOSUB L900
50 'L50;LINEA[0,0,639,147,0,4,2];LINEA[0,149,639
,199,0,6,2]
60 'LINEA[296,100,300,148,0,1,2];LINEA[339,100,3
43,148,0,1,2];LINEA[300,100,339,120,0,3,2]
70 'MT=0;FT=0
80 'MX=18;MXX=18;FX=20;FXX=20;BX=20;BXX=20;BYY=9
0;T=1;L=10;M=40
90 'FPUT[40+MX,100,6,48,1,8]
100 'FPUT[FX,100,6,48,5,8]
110 '
120 '(*          メイン ルーチン          *)
130 '
140 '
150 'L150;LINEA[200,169,439,199,0,0,2]
160 'SYMBOL[219,170,48+(FT>9),FT+48-10*(FT>9),58
,48+(MT>9),MT+48-10*(MT>9),5,4,1,0,0]
170 'IF MT<=14 AND FT<=14 THEN GOTO L210;FI
180 'SYMBOL[0,50,'◆','ゲ','゛','ー','ム',8,4,3,0,0];
SYMBOL[320,50,'・','セ','ッ','ト','◆',8,4,3,0,0];SYM
BOL[0,100,'*',' ','キ','ー','ヲ',5,4,1,0,0];SYMBOL[
200,100,'オ','シ','テ','ク','ダ',5,4,1,0,0];SYMBOL[40
0,100,'゛','サ','イ',' ','*',5,4,1,0,0];KS=INKEY[];
POKE $68FF,0
190 'L190;KS=INKEY[]
200 'IF KS>0 THEN GOTO L50 ELSE GOTO L190;FI
210 'L210;BY=90
220 'GOSUB L810
230 'IF BX<40 AND FS='C' THEN FOR I=0 TO 10000;N
EXT;GOTO L280;FI
240 'IF BX>40 AND MS='C' THEN FOR I=0 TO 10000;N
EXT;GOTO L280;FI
250 'L250;KS=0;KS=INKEY[]
260 'IF KS=6 AND BX>40 GOTO L280 ;FI
270 'IF KS=4 AND BX<40 GOTO L280 ELSE GOTO L250;
FI
280 'L280;FOR I=0 TO 500;BEEP1[];BEEP0[];NEXT;IF
 BX>40 THEN FPUT[40+MX,92,6,12,9,8] ELSE FPUT[FX
,92,6,12,10,8];FI
290 'TX=BX;GOSUB L840
300 '
310 '(*       タマ イドウ       *)
320 'L320;FOR I=0 TO 700;NEXT;BX=BX+T;BY=(BX-M)*
(BX-M)/L;IF BY>=90 THEN BY=90;FPUT[BXX,BYY,1,4,1
5,4];BXX=BX;BYY=BY;FPUT[BX,BY,1,4,11+R,2];FOR I=
0 TO 5000;NEXT;FI
330 'IF T<0 GOTO L470;FI
340 '(*        ミギ        *)
350 'IF MS<>'C' THEN GOTO L390;FI
360 'IF BY=90 THEN K=8;GOTO L400;FI
370 'IF BX<40 OR RND(10)>3 THEN GOTO L430;FI
380 'K=7*(BX<MX+40)+9*(BX>MX+43);GOTO L400
390 'L390;K=INKEY[]
400 'L400;IF K>6 THEN MX=MX+2*(K-8);MX=MX-2*(MX>
34)+2*(MX<3);GOTO L410 ELSE GOTO L430;FI
410 'L410;IF K=8 THEN FOR I=0 TO 1000;NEXT;IF BX
>40 AND BY=90 THEN FPUT[40+MX,92,6,12,9,8];X=MX+
40;GOTO L590;FI;GOTO L670;FI
420 'EX=MXX+40+4*(K=7);FPUT[EX,100,2,48,15,4];MX
X=MX;FPUT[40+MX,100,6,48,1,8]
430 'L430;GOTO L670
440 '
450 '(*      ヒダリ      *)
460 '
470 'L470; IF FS<>'C' THEN GOTO L510;FI
480 'IF BY=90 THEN K=2;GOTO L520;FI
490 'IF BX>39 OR RND(10)>3 THEN GOTO L550;FI
500 'K=3*(BX>5+FX)+(BX<FX+3);GOTO L520
510 'L510; K=INKEY[]
520 'L520; IF K=0 OR K>3 THEN GOTO L430 ELSE FX=
FX+2*(K-2);FX=FX-2*(FX>31)+2*(FX<0);GOTO L530 EL
SE GOTO L550;FI
530 'L530;IF K=2 THEN FOR I=0 TO 1000;NEXT;IF BY
=90 AND BX<40 THEN FPUT[FX,92,6,12,10,8];X=FX;GO
TO L590;FI;GOTO L670;FI
540 'EX=FXX+4*(K=1);FPUT[EX,100,2,48,15,4];FXX=F
X;FPUT[FX,100,6;48,5,8]
550 'L550;GOTO L670
560 '
570 '(*       タマ ヲ ウツ       *)
580 '
590 'L590;FOR I=0 TO 200;BEEP1[];BEEP0[];NEXT;IF
 T*(X-39)<0 OR BY<90 THEN GOSUB L840;GOTO L670;F
I
600 'IF X<BX-6 OR X>BX+1 THEN GOSUB L840;GOTO L6
70;FI
610 'FPUT[BXX,BYY,1,4,15,4];BXX=BX;BYY=BY
620 'GOSUB L840;FPUT[BX,BY,1,4,11+R,8]
630 'GOSUB L850;T=-T;BY=100;GOTO L320
640 '
650 '(*      タマ  ヒョウジ      *)
660 '
670 'L670;GOSUB L810;IF BY<90 OR (T*(BX-40))<0 G
OTO L320;FI
680 '
690 '(*     ミス  ショット      *)
700 '
710 'BEEP1[];FOR I=0 TO 500;NEXT;BEEP0[]
720 'BY=140'
730 'FPUT[BXX,BYY,1,4,15,4];BXX=BX;BYY=BY
740 'R=(BY<20)*(1+2*(T<0))+(BY>=20)*(1+SGN((40-B
X)*T))
750 'FPUT[BX,BY,1,4,11+R,8]
760 'IF BX>39 THEN FT=FT+(TX<39) ELSE MT=MT+(TX>
40);FI
770 'BX=80-BX;GOSUB L850
780 'FOR I=0 TO 1000;NEXT;GOTO L150
790 '
800 ' (*   サブ   *)
810 'L810;FPUT[BXX,BYY,1,4,15,4];BXX=BX;BYY=BY
820 'R=(BY<20)*(1+2*(T<0))+(BY>=20)*(1+SGN((40-B
X)*T))
830 'FPUT[BX,BY,1,4,11+R,8];RETURN
840 'L840;FOR N=0 TO 1000;NEXT;IF BX<40 THEN FPU
T[FX,92,6,10,15,4];FPUT[FX,100,6,48,5,8] ELSE FP
UT[40+MX,92,6,10,15,4];FPUT[40+MX,100,6,48,1,8];
FI;RETURN
850 'L850;IF BX>40 THEN X=MX+44 ELSE X=FX;FI;M=X
+54+RND(25)-54*(BX>40);M=M/2;L=X-M;L=L*L/100;BX=
X;POKE $68FF,0;RETURN
860 '
870 '(*   ヨミコミ   *)
880 '
890 '
900 'L900;
910 'FPRO[]
920 'FOR I=0 TO 9;ADDRES=$6900+$100*I;FGET[ADDRE
S,I+1];NEXT
930 'FGET[$7300,15]
940 'FOR I=0 TO 3;ADDRES=$7400+12*I;FGET[ADDRES,
I+11];NEXT
950 'RETURN
10000 '
10010 '(*   サブシステム   *)
10020 'BEEP1;POKE $FD03,$81;RETURN
10030 'BEEP0;POKE $FD03,$80;RETURN
10040 'CLS;PRINT CHR$(12);RETURN
10050 'LINEA;HLT[];CPOKE ADDR,$15,%6,%5,%1#,%2#,
%3#,%4#,%7;SUB[];RETURN
10060 'SYMBOL;HLT[];CPOKE ADDR,25,%10,%12,%11,%8
,%9,%1#,%2#,5,%3,%4,%5,%6,%7;SUB[];RETURN
10069 'INKEY;HLT[];CPOKE  ADDR,$3F,'Y','A','M','
A','U','C','H','I',$93,$D38F#,$90,$B6,$D401#,$B7
,$D382#,$B6,$D40A#,$39;SUB[];HLT[];KD=PEEK($FC82
);SUB[]
10070 'KY=PEEK($68FF);IF KD>0 THEN KY=KD;FI;IF K
Y<48 OR KY>57 THEN KY=48;FI;POKE $68FF,KY;KY=KY-
48;RETURN
10080 'HLT;CODE $B6,$FD,$05,$2B,$FB;POKE $FD05,$
80;CODE $B6,$FD,$05,$2A,$FB;ADDR=$FC82;RETURN
10090 'SUB;POKE $FD05,0;RETURN
40000 '(* *************  F GET ********* *)
40010 'FGET;A=$C000+$100*%2;ADDRES=%1;FOR M=0 TO
 3;HLT[];CPOKE ADDR,$3F,'Y','A','M','A','U','C',
'H','I',$91,$D393#,A#,$0040#,$90;FOR N=0 TO $3F;
ADDR:N)=ADDRES:N);NEXT;SUB[];ADDRES=ADDRES+$40;A
=A+$40;NEXT;RETURN
40020 '(* *********  F PUT ************* *)
40030 'FPUT;A=%1+80*%2;HLT[];CPOKE ADDR,$3F,'Y',
'A','M','A','U','C','H','I',$91,$D396#,$C000#,$0
006#,$93,$C00B#,$90,A#,%3,%4,%5,%6;SUB[];RETURN
40040 '
40050 '(* *** F PUT PROGRAM TRANSPORT  *** *)
40060 'FPRO;DATA=$680B
```

リスト1

```
40070 'CPOKE DATA,$34,$37,$B6,$D4,$09
40080 'CPOKE DATA,$CC,$C0,$00,$AB,$8C,$EE,$ED,$8
C
40090 'CPOKE DATA,$ED,$1F,$01,$6F,$8C,$EC,$1A,$5
1
40100 'CPOKE DATA,$69,$8C,$E7,$A6,$8C,$E4,$81,$0
8
40110 'CPOKE DATA,$26,$02,$35,$B7,$A6,$8C,$D6,$8
1
40120 'CPOKE DATA,$08,$24,$0F,$AE,$8C,$D0,$A5,$8
C
40130 'CPOKE DATA,$D1,$26,$03,$4F,$10,$21,$86,$0
5
40140 'CPOKE DATA,$10,$21,$86,$07,$A7,$8C,$16,$E
6
40150 'CPOKE DATA,$8C,$B8,$4F,$E3,$8C,$B2,$1F,$0
2
40160 'CPOKE DATA,$EC,$8C,$AF,$40,$ED,$8C,$B1,$E
6
40170 'CPOKE DATA,$8C,$AE,$A6,$80,$20,$0B,$43,$A
4
40180 'CPOKE DATA,$A5,$10,$21,$AA,$A5,$A7,$A5,$5
C
40190 'CPOKE DATA,$2D,$F0,$31,$A8,$50,$6A,$8C,$9
9
40200 'CPOKE DATA,$26,$E5,$A6,$8C,$8B,$8B,$40,$A
7
40210 'CPOKE DATA,$8C,$86,$20,$A4
40220 'FGET[$6800,0];RETURN
```

リスト2　バトミントン・ゲーム　マシン語リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6900 14 00 00 00 00 00 55 00 00 00 00 00 55 00 FF 00 :BD
6910 00 00 55 03 00 06 00 00 14 0C 00 01 80 00 2A 30 :59
6920 00 00 60 00 2A 19 F7 C0 60 00 55 09 36 60 60 00 :0E
6930 55 03 36 69 60 00 55 00 A5 55 60 00 2A 01 45 54 :CA
6940 60 00 55 41 40 50 40 00 55 50 55 5C 40 00 15 57 :C8
6950 15 40 00 00 05 5F 15 0F E0 00 01 FB 80 1F F8 00 :50
6960 00 FB 92 7F FE 00 00 3B F3 FF FC 00 00 0B F3 FF :30
6970 FA 80 00 03 F3 FC EA A0 00 03 F3 FF 15 50 00 01 :51
6980 FF FF CA A8 00 00 FF FF C1 54 00 00 7F FF C5 54 :1A
6990 00 00 7F FF 8A A8 00 00 7F FF AA A0 00 00 24 92 :2E
69A0 2A 80 00 00 7F FA AA 00 00 00 7F F5 50 00 00 00 :91
69B0 FF F5 40 00 00 01 DF FF C0 00 00 01 6F FF C0 00 :02
69C0 00 05 4F FF 80 00 00 15 52 AA 80 00 00 2A A9 55 :8C
69D0 40 00 00 55 50 55 40 00 00 AA A0 55 A0 00 01 55 :0F
69E0 00 2A A0 00 00 55 40 15 50 00 00 55 A0 0A A0 00 :63
69F0 00 15 80 02 8F 00 00 05 38 00 3F C0 00 00 FC 00 :5E
---------------------------------------------------------
Sum: 40 76 CA 2C 28 17 E8 D7 1B 5A 82 60 88 0D BD 6B :BE


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6A00 3F E0 00 01 FE 00 0F C8 00 00 79 00 02 BC 00 00 :2C
6A10 0B C0 01 FC 00 00 3F 80 07 F0 00 00 FC 00 0F C0 :49
6A20 3C 00 00 FF FF C0 FF FF FF D5 55 F0 FF 00 00 FF :0F
6A30 FF C0 FF 00 00 00 00 00 3C 00 00 00 00 7E 00 :78
6A40 00 00 00 00 7F 00 0C 00 00 00 FF 00 0C 30 00 00 :C6
6A50 FF 00 0C 3B 00 00 FF 01 E7 FF 00 00 7E 03 CF FC :78
6A60 00 00 7F C3 F5 F8 00 00 7F F0 FF F8 00 00 3F FF :D3
6A70 FF C8 00 00 0F FF D5 7F E0 00 03 FB F7 FF F8 00 :F5
6A80 00 FB FF FF FE 00 00 3B FF FF FE 00 00 0B FF FF :37
6A90 FF 80 00 03 FF FC FF E0 00 03 FF FF 3F F0 00 01 :8D
6AA0 FF FF CF F8 00 00 FF FF C3 FC 00 00 7F FF CF FC :CB
6AB0 00 00 7F FF 9F F8 00 00 7F FF BF E0 00 00 00 00 :32
6AC0 7F 80 00 00 7F FB FE 00 00 00 7F F7 F0 00 00 00 :DD
6AD0 FF FB C0 00 00 01 DF FF C0 00 00 01 EF FF C0 00 :08
6AE0 00 07 EF FF 80 00 00 1F FB FF 80 00 00 3F FD FF :49
6AF0 C0 00 00 7F F8 FF C0 00 00 FF E0 7F E0 00 01 FF :34
---------------------------------------------------------
Sum: BF 24 87 71 13 A6 C8 FF 84 AF 6A 39 FB 26 1F B4 :25


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6B00 80 3F E0 00 00 FF C0 1F F8 00 00 7F E0 0F FC 00 :DF
6B10 00 1F F0 03 FF 00 00 07 F8 00 FF C0 00 07 FC 00 :D2
6B20 3F E0 00 01 FE 00 0F C8 00 00 79 00 02 BC 00 00 :2C
6B30 0B C0 01 FC 00 00 3F 80 07 F0 00 00 FC 00 0F C0 :49
6B40 D7 FF FF 00 00 3F 55 00 00 3F FF 0F 55 FF 00 00 :0A
6B50 00 3F 55 FC 00 01 FF FF D7 F0 00 00 7F FF AB C0 :3F
6B60 00 00 1F FF AB E0 0C 00 1F FF 55 F1 36 60 1F FF :CD
6B70 55 FF 36 69 1F FF 55 FE A5 55 1F FF AB FD 45 54 :BD
6B80 1F FF 55 7D 40 50 3F FF 55 5F 55 54 7F FF D5 57 :C5
6B90 15 47 FF FF F5 5F 15 0F FF FF FD FB 80 1F FF FF :65
6BA0 FF FB 92 7F FF FF FF FB F3 FF FD FF FF FB F3 FF :DD
6BB0 FA FF FF FF F3 FC EA BF FF FF F3 FF 15 5F FF FF :F1
6BC0 FF FF FA AF FF FF FF FF FD 57 FF FF 7F FF C5 57 :8F
6BD0 FF FF FF FF 8A AF FF FF FF FF AA BF FF FF A4 92 :CE
6BE0 2A FF FF FF FF FA AB FF FF FF FF F5 5F FF FF FF :18
6BF0 FF F5 7F FF FF FF DF FF FF FF FF FF 6F FF FF FF :B6
---------------------------------------------------------
Sum: 4A 6D D6 0A 75 6F 88 2F D2 23 D4 3D F2 A1 43 0E :1C


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6C00 FF FD 4F FF FF FF FF F5 52 AA FF FF FF EA A9 55 :1D
6C10 7F FF FF D5 57 55 7F FF FF AA BF D5 BF FF FF 55 :CB
6C20 7F EA BF FF FF 55 7F F5 5F FF FF D5 BF FA A3 FF :7C
6C30 FF F5 8F FE 8F FF FF FD 3F FF 3F FF FF FC FF FF :80
6C40 FF FF FF FF FF FF FF CF FF FF F9 FF FE BF FF FF :79
6C50 CB FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :BC
6C60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6C70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6C80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6C90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6CA0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6CB0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6CC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6CD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6CE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
6CF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
---------------------------------------------------------
Sum: C6 D9 9A CF E2 A6 FA B4 ED 50 F4 A6 79 9D 48 A6 :19


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6D00 00 00 00 00 00 14 00 00 00 00 00 54 00 00 00 FF :67
6D10 00 55 00 00 0F 00 F0 54 00 E0 F0 00 0C 14 01 0C :A5
6D20 00 01 42 2A 02 0C 00 55 44 55 02 0C 00 5C C4 55 :EC
6D30 02 11 00 5C E0 55 02 11 00 55 40 55 02 21 00 55 :19
6D40 50 55 02 21 00 AA 01 55 02 40 55 55 45 54 01 C0 :0E
6D50 07 FF E5 54 00 00 3F FF F5 50 00 00 FF FF FE 80 :3E
6D60 00 1F 7F FF FF 00 00 FF 7F FF FC 00 02 BF 7F FD :52
6D70 40 00 05 5F 7F FF C0 00 15 57 7F FF 00 00 15 54 :35
6D80 7F FE 00 00 15 20 1F FE 00 00 15 80 1F FC 00 00 :7F
6D90 15 80 09 24 00 00 15 80 3F FC 00 00 15 00 DD B4 :38
6DA0 00 00 05 03 BB B4 00 00 1A 03 BB 77 00 00 2A 87 :77
6DB0 77 75 C0 00 1A 07 77 6A 70 00 0A 0F 7B 6F D0 00 :F1
6DC0 00 0F FF FD 54 00 00 01 54 55 55 00 00 01 55 15 :C9
6DD0 55 40 00 01 55 05 55 40 00 01 55 00 AA 80 00 05 :0A
6DE0 55 00 2A 80 00 05 AA 00 2A 80 00 22 A8 00 2A 80 :CC
6DF0 00 74 50 00 00 00 00 FF 00 00 7F 00 00 BF 00 00 :01
---------------------------------------------------------
Sum: 4E 90 F4 FE 02 03 9C 35 16 45 05 31 55 4E AE 1B :A3


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6E00 7F 00 07 AC 00 00 7F 00 0F E0 00 00 2A 00 07 F8 :C9
6E10 00 00 7F 00 01 FC 00 00 7F E0 00 3C 00 00 7F F0 :86
6E20 03 FF FF 00 00 3C 0F 55 57 FF FF FE 03 FF FF 00 :F5
6E30 00 FF 00 00 00 00 00 7E 00 00 00 00 00 3C 00 00 :B9
6E40 00 03 C0 7E 00 00 00 F5 60 FF 00 00 00 F0 20 FF :A4
6E50 00 00 00 78 20 FF 00 00 00 7F C0 FF 00 00 00 7F :54
6E60 F0 FF 00 00 00 FF C1 7F 00 00 00 BF CF FE 00 00 :BA
6E70 07 FB FF FE 00 00 3F FF FF F0 00 00 FF FF FF C0 :E9
6E80 00 1F 7F FF FF 00 00 FF 7F FF FC 00 03 FF 7F FD :93
6E90 40 00 0F FF 7F FF C0 00 3F FF 7F FF 00 00 3F FC :83
6EA0 7F FE 00 00 3F 60 1F FE 00 00 3F 80 1F FC 00 00 :13
6EB0 3F 80 00 00 00 00 3F 80 3F FC 00 00 1F 00 DD B4 :69
6EC0 00 00 0F 03 BB B4 00 00 1F 03 BB 77 00 00 3F 87 :9B
6ED0 77 75 C0 00 1E 07 77 6A 70 00 0E 0F 7B 6F F0 00 :19
6EE0 00 0F FF FF FC 00 00 03 FC FF FF 00 00 03 FF 3F :47
6EF0 FF C0 00 03 FF 0F FF C0 00 03 FF 01 FF 80 00 07 :18
---------------------------------------------------------
Sum: ED DC A0 A3 B2 5F 22 F0 CC 2C 40 FE B6 15 6D A0 :3D


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
6F00 FF 00 3F 80 00 1F FE 00 7F 80 00 3F F8 00 2F 80 :C0
6F10 00 7F F0 00 7F 80 00 FF C0 00 7F 00 00 BF 00 00 :6B
6F20 7F 00 07 AC 00 00 7F 00 0F E0 00 00 2A 00 07 F8 :C9
6F30 00 00 7F 00 01 FC 00 00 7F E0 00 3C 00 00 7F F0 :86
6F40 FC 00 00 FF FF D7 F0 FF FC 00 00 55 FC 00 00 00 :0D
6F50 FF 55 FF FF F0 00 0F D5 FF 1F 00 00 03 D7 FE 0C :28
6F60 00 01 43 AB FC 04 00 55 43 55 FC 04 00 5C C7 55 :54
6F70 FC 0C 00 5C FF 55 FC 0C 00 55 7F 55 FC 1C 00 55 :56
6F80 5F 55 FC 1E 00 AA 3F 55 FC 3F D5 55 75 57 FE 3F :7A
6F90 F8 03 E5 57 FF FF FF FF F5 5F FF FF FF FF FE BF :40
6FA0 FF FF 7F FF FF FF FF FF 7F FF FF FF FE BF 7F FD :2D
6FB0 7F FF F5 5F 7F FF FF FF D5 57 7F FF FF FF D5 57 :22
6FC0 FF FF FF FF D5 3F FF FF FF FF D5 FF FF FF FF FF :DC
6FD0 D5 FF E9 27 FF FF D5 FF FF FF FF FF D5 FF DD B7 :1A
6FE0 FF FF F5 FF BB B7 FF FF DB FF BB 77 FF FF AA FF :15
6FF0 77 75 FF FF CB FF 77 6A 7F FF EB FF 7B 6F DF FF :C5
---------------------------------------------------------
Sum: 94 A9 28 28 41 66 FE ED A8 F9 C6 EF DC 8E 2F 24 :32


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7000 FF FF FF FD 57 FF FF FD 54 55 55 FF FF FD 55 15 :AF
7010 55 7F FF FD 55 F5 55 7F FF FD 55 FE AA FF FF FD :E2
7020 55 FF 2A FF FF E5 AB FF AA FF FF E2 AF FF AA FF :EC
7030 FF F4 5F FF 80 7F FF FF 3F FF FF FF FE BF FF FF :45
7040 FF FF FF AF FF FF FF FF FF FF FF FF AA FF FF FF :4B
```

リスト2

```
7050 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
70A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
70B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
70C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
70D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
70E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
70F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-------------------------------------------------------
Sum: A6 6F 85 A6 29 56 FC 78 3A 4E A6 DC FF B8 FB 0E :FD


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7130 14 00 00 00 00 00 55 00 00 00 00 00 55 00 00 00 :BE
7140 00 00 55 00 00 00 00 00 00 00 00 00 00 00 00 00 :55
7150 00 03 FF C0 00 00 00 3F 00 C0 00 00 03 F0 0F C0 :83
7160 00 00 0F 00 FC 00 00 00 3F FF C0 00 00 03 FC 00 :08
7170 00 00 00 3F 00 00 00 00 3F F0 00 00 00 00 FF 00 :6D
7180 00 00 00 00 FF 00 00 00 00 00 FF 00 00 00 00 00 :FE
7190 FF FF FF FF FF FF FF FF FF FC 00 3F FF FF FF C0 :EF
71A0 FF 3F FF FF FC 0F F0 3F FF FF F0 FF 03 FF FF FF :63
71B0 C0 00 3F FF FF FC 03 FF FF FF FF C0 FF FF FF FF :B4
71C0 D4 0F FF FF FF FF 55 FF FF FF FF FF 55 FF 00 FF :82
71D0 FF FF 55 FC 00 00 FF FF 00 00 00 00 00 00 00 00 :4D
71E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
71F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-------------------------------------------------------
Sum: A5 4F F4 F7 F4 09 9B 7A 7A A8 AD FD AE EF 07 7D :DE


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
7230 00 00 00 00 00 14 00 00 00 00 00 55 00 00 00 00 :69
7240 00 55 00 00 00 00 00 14 00 00 00 00 00 00 03 FF :6B
```

```
7250 C0 00 00 00 03 00 FC 00 00 00 03 F0 0F C0 00 00 :81
7260 00 3F 00 F0 00 00 00 03 FF FC 00 00 00 00 00 3F :6C
7270 C0 00 00 00 00 00 FC 00 00 00 00 00 0F FC 00 00 :C7
7280 00 00 00 7E 00 00 00 00 00 FF 00 00 00 00 00 3C :B9
7290 FF FF FF FF FF FF FC 00 3F FF FF FF FC FF 03 FF :2F
72A0 FF FF FC 0F F0 3F FF FF FF C0 FF 0F FF FF FF FC :FC
72B0 00 03 FF FF FF FF FF C0 3F FF FF FF FF FF 03 FF :FA
72C0 FF FF FF FF F0 17 FF FF FF FF FF D5 FF FF FF 00 :D0
72D0 FF 55 FF FF F0 00 0F D7 00 00 00 00 00 00 00 00 :28
72E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
72F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :00
-------------------------------------------------------
Sum: 7C E9 F8 79 D1 68 00 AC 7B B8 FF 27 17 B8 07 74 :5E


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7300 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7310 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7320 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7330 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7340 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7350 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7360 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7370 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7380 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
7390 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
73A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
73B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
73C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
73D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
73E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
73F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :F0
-------------------------------------------------------
Sum: F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 :00


Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F Sum
7400 FF 7E 3C 00 FF 7E 00 3C FF FF C3 C3 00 F0 FC F0 :D2
7410 00 F0 F3 F0 FF FF F0 FF 00 3C 7E FF 18 00 7E FF :0E
7420 C3 C3 FF FF 00 0F 3F 0F 00 0F CF 0F FF FF 0F FF :DA
-------------------------------------------------------
Sum: C2 31 2E EF FE 8C 2F 4A FF 4A 10 D1 17 EF 89 EE :BA
```

RANDOM BOX

FM-8用プログラムのFM-11への移植

●浅見　薫

FM-11にはまだソフトが出ていないので大方は7/8用のものを利用していると思います．オールBASICでROMルーチンをCALLしていないソフトはSCREEN文を追加すれば走ります．

例　FM-8活用研究〝マージャン〟など．

また一見マシン語に見えますが，グラフィック情報としてマシン語領域にデータを持っているソフトもPUT，GUT文を使ってるものであれば走ります．参考例としてFM-8活用研究の〝花札〟をFM-11用に改造してみます．

まず，もとのソフトはグラフィック・データを&H34F0～&H7FFFまで持っており，11は&H7C000～&H8000までテキスト・ウインドウなのでデータをロードするときに&H1000だけシフトさせてLOADM〝DATA〟，一&H1000でロードします．モニタでシフトしたことを確認して，次にBASICの&Hがついたところをすべて&H1000引いて書き換えます．

次にINPUT文の前がSYMBOL文なのでPRINT文に書き換えます．これでFM-11でもFM-8のソフトが走ります．

改造例(あくまでも1例です．もとのプログラムで&H37E6を&H27E6に&H1000引いて書き換える)

```
0 CLS:CLEAR,&H24F2:LOADM"DATA":DEFINTA-Z:DEFUSR0=&H27E6:DEFUSR1=&H2966:DEFUSR2=&
H2572:WIDTH40,25:COLOR7,4


920 COL=6:XX=208:YY=0:FD=24:PT=192:AD=&H29E0:GOSUB10000:XX=336:PT=96:AD=&H2AA0:G
OSUB10000:XX=400:AD=&H2AA0:GOSUB10000
930 COL=2:XX=8:YY=8:FD=16:PT=96:AD=&H2B00:GOSUB10000:YY=72:AD=&H2B60:GOSUB10000
940 LOCATE 1,23:PRINT"1  2   3   4  5  6  7   8"


2020 LOCATE0,24:PRINT"ｼｮｳﾌﾞ= 0 ,ｺｲｺｲ = 1";:LOCATE14,24:INPUTSM:CLS4


7570 LINE(0,0)-(639,199),PSET,0,BF


10000 POKE &H29D8,(VARPTR(A(0))¥ 256):POKE &H29D9,(VARPTR(A(0)) MOD 256):POKE &H
29D7,PT:SS=USR1(AD):LINE(XX,YY)-(XX+16*PT/FD-1,YY+FD-1),PSET,0,BF:PUT@(XX,YY)-(X
X+16*PT/FD-1,YY+FD-1),A,PSET,COL:RETURN
20000 POKE &H29D8,(VARPTR(A(0))¥ 256):POKE &H29D9,(VARPTR(A(0)) MOD 256):SS=USR0
(AD):PUT@A(XX,YY)-(XX+63,YY+46),A,PSET:BEEP1:BEEP0:RETURN
```

資料編

FM-11 サブシステム・コマンド一覧 …… 210
FM-11 ハード仕様 …… 219
FM-11 FM-11回路図 …… 246
FM-11 実用ソフト・カタログ …… 307
FM-7/8 活用研究正誤表 …… 334

FM-11

Display Subsystem コマンド

コンソール・コマンド

CONSOLE INIT

コンソール画面の各種制御パラメータの設定を行ないます.

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't care
2	C	コマンド・コード	$01
3	AT	アトリビュート	[ERS]≠0のときに有効
4	NC	画面桁数	40または80
5	NL	画面行数	20または25
6	SL	スクロール開始行番号	0～[NL]
7	NS	スクロール行数	0～([NL]－[SL])
8	FD	ファンクション・キー表示フラグ	[FD]＝0で表示しない [FD]≠0で表示する
9	ERS	コンソール画面消去フラグ	[ERS]≠0で消去しない [ERS]≠0で消去する

ERASE CONSOLE VRAM

コンソール画面(文字VRAM)を消去します.

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't care
2	C	コマンド・コード	$02
3	W	消去範囲	[W]＝0：全画面 [W]＝1：スクロール・モード画面 [W]＝2：ページ・モード1画面 [W]＝3：ページ・モード2画面
4	AT	アトリビュート	アトリビュート・コード

PUT STRING

文字列をコンソール画面に出力します. 文字列には画面に直接表示される文字の他に, 制御用のオーダーコードやエスケープ・シーケンス・コードも含まれます.

コマンド＆パラメータ

相対値	記号	内容	値
0	—		Don't care
1		継続フラグ・ビット(MSB)	[CN]＝1のときにデータが継続することを示す
2	C	コマンド・コード	$03
3	N	文字列のバイト数	1～124
4～127	String	出力文字列	出力文字列(オーダーを含む)

GET

文字列をコンソール画面に表示した後に, ローカル・エディット・モードになり, オペレータのキー入力を受け付けます. ローカル・エディットが終了すると, オペレータによって変更されたフィールドのうちで画面最上段に位置するフィールドを読取ります. 変更されたフィールドのない場合にはカーソル位置を含むフィールドを読み取ります.

コマンド＆パラメータ

相対値	記号	内容	値
0	—		Don't Care
1	CN	継続フラグ・ビット(MSB)	[CN]＝1のときにデータが継続することを示す.
2	C	コマンド・コード	$04
3	N	文字列のバイト数	1～124
4～127	String	出力文字列	出力文字列(オーダを含む)

復帰情報

相対値	記号	内容	値
0	E	エラーコード	[E]≠0のときエラーあり
1	CN	継続フラグ・ビット(MSB)	[CN]＝1のときにデータが継続することを示す.
2	—		$00
3	K	終端キー種類	0：RETURN 1：CTRL＋C または CTRL＋X
4	N	文字列のバイト数	TEXTのバイト数＋3
5	—	Set Cursorオーダーコード	$12
6	X	フィールド先頭X座標	0～79(コンソール座標)
7	Y	フィールド先頭Y座標	0～24(コンソール座標)
8～127	TEXT	フィールド文字列	文字列

エラー内容については, エラーコード表を参照.

GETC

GETコマンドによって変更されたフィールドのうち, まだ読み込まれていないフィールド(未転送フィールド)の読み込みをします. スクリーン・エディタなどのように複数のフィールドを対象とするプログラムの場合に使用します.

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$05

復帰情報

相対値	記号	内容	値
0	E	エラーコード	[E]≠0のときエラーあり
1	CN	継続フラグ・ビット(MSB)	[CN]＝1のときにデータが継続することを示す.
2	—		$00
3	K	終端キー種類	0：RETURN 1：CTRL＋C または CTRL＋X
4	N	文字列のバイト数	TEXTのバイト数＋3
5	—	Set Cursor オーダーコード	$12
6	X	フィールド先頭X座標	0～79(コンソール座標)
7	Y	フィールド先頭Y座標	0～24(コンソール座標)
8～127	TEXT	フィールド文字列	文字列

エラー内容については, エラーコード表を参照.

GET CHARACTER BLOCK 1

対角線座標(X_1, Y_1)，(X_2, Y_2)によって示される枠内の文字コードを読み取ります.

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$06
3	X_1	X_1座標	0～79(コンソール座標)
4	Y_1	Y_1座標	0～24(コンソール座標)
5	X_2	X_2座標	0～79(コンソール座標)
6	Y_2	Y_2座標	0～24(コンソール座標)

X_1, Y_1, X_2, Y_2によって画面上の座標を指定します.

復帰情報

相対値	記号	内容	値
0	E	エラーコード	[E]≠0のときエラーあり
1	CN	継続フラグ・ビット(MSB)	[CN]＝1のときにデータが継続することを示す.
2	—		$00
3	B	今回転送バイト数	1～124
4～127	Cn	文字コード	文字コード

PUT CHARACTER BLOCK 1

対角線座標(X_1, Y_1)，(X_2, Y_2)によって示される枠内へ文字を表示します.

コマンド＆パラメータ

相対値	記号	内容	値
0	—		Don't Care
1	CN	継続フラグ・ビット(MSB)	[CN]＝1のときにデータが継続することを示す.
2	C	コマンド・コード	$07
3	X_1	X_1座標	0～79(コンソール座標)
4	Y_1	Y_1座標	0～24(コンソール座標)
5	X_2	X_2座標	0～79(コンソール座標)
6	Y_2	Y_2座標	0～24(コンソール座標)
7	AT	アトリビュート	アトリビュート文字
8	B	今回転送バイト数	1～120
9～127	Cn	文字コード	文字コード

GET CHARACTER BLOCK 2

対角線座標(X_1, Y_1)，(X_2, Y_2)によって示される枠内のアトリビュートと文字コードを読み取ります.

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$08
3	X_1	X_1座標	0～79(コンソール座標)
4	Y_1	Y_1座標	0～24(コンソール座標)
5	X_2	X_2座標	0～79(コンソール座標)
6	Y_2	Y_2座標	0～24(コンソール座標)

X_1, Y_1, X_2, Y_2によって画面上の座標を指定します.

復帰情報

相対値	記号	内容	値
0	E	エラーコード	[E]≠0のときエラーあり
1	CN	継続フラグ・ビット(MSB)	[CN]＝1のときにデータが継続することを示す.
2	—		$00
3	B	今回転送バイト数	2～124
2＋2・n	atn	アトリビュート	アトリビュート
3＋2・n	Cn	文字コード	文字コード

エラー内容についてはエラーコード表を参照.

PUT CHARACTER BLOCK 2

対角線座標(X_1, Y_1)，(X_2, Y_2)によって示される枠内に，アトリビュート付きの文字を表示します.

コマンド＆パラメータ

相対値	記号	内容	値
0	—		Don't care
1	CN	継続フラグ・ビット(MSB)	[CN]＝1のときにデータが継続することを示す.
2	C	コマンド・コード	$09
3	X_1	X_1座標	0～79(コンソール座標)
4	Y_1	Y_1座標	0～24(コンソール座標)
5	X_2	X_2座標	0～79(コンソール座標)
6	Y_2	Y_2座標	0～24(コンソール座標)
7	—		Don't care
8	B	今回転送バイト数	2～119
7＋2・n	atn	アトリビュート	アトリビュート
8＋2・n	Cn	文字コード	文字コード

READ CONSOLE PARAMETER

現在設定されている，コンソール・パラメータの値を読み込みます.

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't cate
2	C	コマンド・コード	$0A

復帰命令

相対値	記号	内容	値
0	E	エラーコード	[E]≠0のときエラーあり
1	—		不定
2	—		$00
3	X	カーソルX座標	0～79(コンソール座標)
4	Y	カーソルY座標	0～24(コンソール座標)
5	AT	アトリビュート	アトリビュート
6	—		不定
7	NC	画面桁数	40または80
8	NL	画面行数	20または25
9	SL	スクロール開始行	0～[NL]
10	NS	スクロール行数	0～([NL]－[SL])
11	FD	ファンクション・キー表示フラグ	[FD]＝0で表示していない [FD]＝1で表示している
12	CF	コンソール制御フラグ	次表参照

エラー内容については15.1 エラーコード表を参照.

BIT	FLAG NAME
0	Cursor Display
1	Order Operation
2	TAB Operation
3	(reserve)
4	Put Wait
5	New Line
6	Hardware Scroll
7	Order Select
8	Clear Page
9	Field Expantion
10	Enable CLS
11	Enable Light Pen Attention
12	Enable Key Attention
13	Auto Line Feed
14	With Graphic School
15	(reserve)

TAB SET

タブ位置を設定します.

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$0B
3～12	TAB	TAB停止位置	80 bit ビット・イメージ

CONSOLE CONTROL

コンソール制御フラグ（16ビット）を制御して，種々のコンソール動作を選択します.

コマンド＆パラメータ

相対値	記 号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$0C
3,4	CF	コンソール制御フラグ	次表参照

コンソール制御フラグ（CF）は，16ビットのフラグ群であり，種々のコンソール動作の選択に用いられます．次の表にその内容を示します．

ビット	フラグ名	内　　容
0	Cursor Display	ON(1)のときにカーソル表示
1	Order Operation	ON(1)のときにオーダ動作をする
2	TAB Operation	ON(1)のときにタブ動作をする
3	リザーブ	
4	Put Wait	ON(1)のときにESCキーの入力によって表示動作を一時停止する．
5	New Line	ON(1)のときにCRオーダ，LFオーダ，それぞれがCR，LF両方の動作を同時に行なう．
6	Hard Ware Scroll	ON(1)のときに全画面スクロールでハードウェア・スクロールを行なう．
7	Order Select	(1)のときにF-BASIC用オーダ，(1)のときにADM-3A用オーダ
8	Clear Page	ON(1)のときに，ページ・モード画面で最終行から，先頭行へ表示が移るときに画面を消去する．
9	Field Expansion	ON(1)のときに，ローカル・エディット・モード時のフィールドの併合，拡張を許す．
10	Enable CLS	ON(1)のときに，ローカル・エディット・モード時のCLSキーによる画面消去を許す．
11	Enable Light Pen Attention	ON(1)のときに，ライトペン割り込みを許可．
12	Enable Key Attention	ON(1)のときに，キー割り込みを許可．
13	Auto Line Feed	ON(1)のときに，画面最右端表示後のオート・ライン・フィード を行なう．
14	With Graphic Scroll	ON(1)のときに，全画面スクロール時にグラフィックもスクロールする．
15	リザーブ	

RESET時は自動的に$06D3がセットされます．また，リザーブの部分は0にしておいてください．

ERASE 2

コンソールVRAMおよびグラフィックVRAM（コンソール画面と同じ範囲）を同時にクリアします．

コマンド＆パラメータ

相対値	記号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$0D
3	W	消去範囲	[W]＝0÷全画面 [W]＝1÷スクロール・モード画面 [W]＝2÷ページ・モード1画面 [W]＝3÷ページ・モード2画面
4	BC	背景色カラーコード	0~7(グラフィック用)
5	AT	アトリビュート	アトリビュート

CLEAR CONSOLE FLAG

コンソール制御フラグの指定されたビットをOFF(0)にします．コンソール制御フラグの特定のビットだけ制御したい場合に用います．

コマンド＆パラメータ

相対値	記 号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$0E
3	BIT	ビット番号	0～15

コンソール制御フラグのうち，ビット番号(BIT)にて示されるビットをOFF(0)にします．以下にコンソール制御フラグのビットの内容表を示します．

BIT	FLAG NAME
0	Cursor Display
1	Order Operation
2	TAB Operation
3	(reserve)
4	Put Wait
5	New Line
6	Hardware Scroll
7	Order Select
8	Clear Page
9	Filed Expantion
10	Enable CLS
11	Enable Light Pen Attention
12	Enable Key Attention
13	Auto Line Feed
14	With Graphic Scrool
15	(reserve)

SET CONSOLE FLAG

コンソール制御フラグの指定されたビットをON(1)にします．コンソール制御フラグの特定のビットだけ制御したい場合に用います．

コマンド＆パラメータ

相対値	記号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$0F
3	BIT	ビット番号	0～15

コンソール制御フラグのうち，ビット番号(BIT)にて示されるビットをON(1)にします．コンソール制御フラグのビット内容はClear Console FlagおよびConsole Controlを参照してください．

COPY CONSOLE VRAM

対解線座標(X_1, Y_1)，(X_2, Y_2)で示される枠内の文字コードまたはアトリビュート(両方も可)を画面の別の部分へ複写します．

コマンド＆パラメータ

相対値	記号	内　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$10
3	X_1	X_1	0～79(コンソール座標)
4	Y_2	Y_1	0～24(コンソール座標)
5	X_2	X_2	0～79(コンソール座標)
6	Y_2	Y_2	0～24(コンソール座標)
7	DX	X座標変位	−79～79
8	DY	Y座標変位	−24～24
9	SL	複写フラグ	1：文字コードのみ複写 2：アトリビュートのみ複写 3：文字コードとアトリビュートを複写

SELECT COMMAND END MODE

Display Subsystemがコマンドの処理を終了したときに，処理の終了をATTENTION割り込みを使ってメインCPUに通知するかどうかの選択をします．

コマンド＆パラメータ

相対値	記号	内　容	値
0,1	—		Don't Care

2	C	コマンド・コード	$11
3	MD	モード	[MD]=0のとき　割り込みをしない [MD]≠0のとき　割り込みをする

モード（MD）には，ATTENTION割り込みを使うかどうかの選択をします．MDの値が0でないときにATTENTION割り込みを使用します．

CHARACTER OUT

コンソール画面に1文字出力します．

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$12
3	CH	出力文字コード	文字コード

出力文字コード（CH）の内容をコンソール画面に出力します．文字コードとして，オーダーコードなどの制御コードも出力できます．

GET LIGHT PEN POSITION

ライトペンが現在示しているコンソール座標を読み取ります．

コマンド＆パラメータ

相対値	記号	内容	値
1,2	—		Don't Care
2	C	コマンド・コード	$19

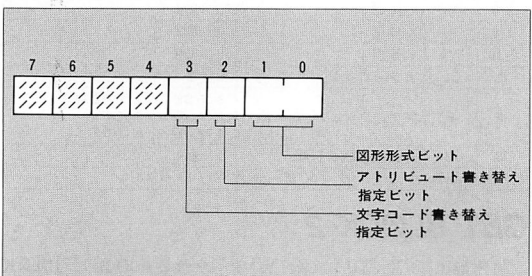
復帰情報

相対値	記号	内容	値
0	E	エラーコード	[E]≠0のときエラーあり
1	—		不定
2	—		$00
3	SW	ライトペン・スイッチ状態	[SW]=$FF　スイッチ ON [SW]=$00　スイッチ OFF
4	X	X座標	0～79(コンソール座標)
5	Y	Y座標	0～24(コンソール座標)

CHARACTER LINE

対角線座標(X_1, Y_1)，(X_2, Y_2)によって示される枠内へアトリビュート付きの文字を表示します．

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$20
3	AT	アトリビュート	アトリビュート
4	CHR	文字コード	文字コード
5	—		Don't Care
6	X_1	X_1座標	0～79(コンソール座標)
7	—		Don't Care
8	Y_1	Y_1座標	0～24(コンソール座標)
9	—		Don't Care
10	X_2	X_2座標	0～79(コンソール座標)
11	—		Don't Care
12	Y_2	Y_2座標	0～24(コンソール座標)
13	B	制御フラグ	下位4ビットデータ

○図形形式ビットは，文字の表示形式を指定します．

00：直線
01：枠(四角形)
10：枠(四角形)内塗りつぶし
00：禁止

○アトリビュート書き替え指定ビット

0：アトリビュート・バッファを書き替える
1：アトリビュート・バッファを書き替えない

○文字コード書き替え指定ビット

0：文字コード・バッファを書き替える
1：文字コード・バッファを書き替えない

グラフィック・コマンド

ファンクション・コード

ファンクション・コードを用いて，描画時のファンクションを指定することができます．ファンクション・コードの一覧表を以下に示します．

コード	機能	作用
0	PSET	指定されたカラーコードで描画
1	PRESET	背景色のカラーコードで描画
2	OR	指定されたカラーコードと画面のカラーコードとをOR（論理和）演算した結果のカラーコードで描画
3	AND	指定されたカラーコードと画面のカラーコードとをAND（論理積）演算した結果のカラーコードで描画
4	XOR	指定されたカラーコードと画面のカラーコードとをXOR（排他的論理和）演算した結果のカラーコードで描画
5	NOT	表示パターンを反転してから，指定したカラーコードで描画

LINE

座標点(X_1, Y_1)，(X_2, Y_2)を結ぶ直線または，(X_1, Y_1)，(X_2, Y_2)を対角線とする四角形を描きます．また四角形の内部を塗りつぶすこともできます．

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$15
3	CL	カラーコード	0～7
4	F	ファンクション・コード	0：PSET　1：PRESET 2：OR　3：AND 4：XOR
5,6	X_1	X_1座標	0～639
7,8	Y_1	Y_1座標	0～399
9,10	X_2	X_2座標	0～639
11,12	Y_2	Y_2座標	0～399
13	B	形状フラグ	0：直　線　1：四角形 2：四角形塗り　3：スタイル付直線 4：スタイル付四角形
14,15	STYLE	ライン・スタイル	16ビット・パターン

CHAIN

指定された座標点(X_n, Y_n)を直線で結びます．座標点は30個まで指定できます．

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$16
3	CL	カラーコード	0～7

4	F	ファンクション・コード	0:PSET 1:PRESET 2:AND 3:OR 4:XOR
5	N	座標点数	2~30
2+4・n 3+4・n	X_n	X_n座標 (1≦n≦〔N〕)	0~639
4+4・n 5+4・n	Y_n	Y_n座標 (1≦n≦〔N〕)	0~399

POINT

指定されたn個の座標点に点を表示します．点の数は最大20個まで指定できます．

コマンド＆パラメータ

相対値	記号	内　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$17
3	N	表示点座標数	1~20
4+6・(n−1) 5+6・(n−1)	X_n	X_n座標 (1≦n≦20)	0~399
6+6・(n−1) 7+6・(n−1)	Y_n	Y_n座標 (1≦n≦20)	0~7
8+6・(n−1)	CL_n	カラーコード	0~7
9+6・(n−1)	F_i	ファンクション・コード	0:PSET 1:PRESET 2:AND 3:OR 4:XOR

PAINT

初期座標点の位置より，指定された境界色に囲まれた部分を，指定色またはタイルストリング・パターンにて塗りつぶします（タイル・ペイント）．

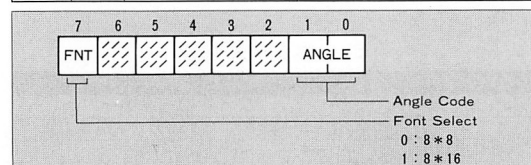
コマンド＆パラメータ

相対値	記号	内　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$18
3,4	X	X座標	0~639
5,6	Y	Y座標	0~399
7	Pc	ペイント・カラーコード	0~7(カラーコード) $FF　タイル・ストリング指定
8	Nc	境界色カラーコードの数	0~8
9~16	Bcn	境界色カラーコード	0~7(カラーコード)
17	Tc	タイル・ストリング組数	1~36
15+3・n 16+3・n 17+3・n	TSn	タイル・ストリング	24ビット・パターン

SYMBOL

文字列を指定座標より指定した角度，大きさで表示します．表示するときに文字を構成するドット以外は変化しません．

コマンド＆パラメータ

相対値	記号	内　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$19
3	CL	カラーコード	0~7
4	F	ファンクション・コード	0:PSET 1:PRESET 2:AND 3:OR 4:XOR 5:NOT
5	A	アングル選択	本文参照
6	W	文字幅倍率	〔W〕=1で8ドット表示
7	H	文字高倍率	〔H〕=1で8または16ドット表示
8,9	X	X座標	0~639
10,11	Y	Y座標	0~399
12	N	文字数	1~80
13~92	String	文字列	文字列

アングル・コード（Angle Code）によって示される0~3の値によって文字の表示アングルおよび表示位置はSYMBOL文のように定義されます．

CHANGE COLOR

対角線座標(X_1, Y_1)，(X_2, Y_2)で示される枠内の，指定された色(カラーコード)のドット色(カラーコード)を変えます．

コマンド＆パラメータ

相対値	記号	内　容	値
0,2	—		Don't Care
2	C	コマンドコード	$1A
3,4	X_1	X_1座標	0~639
5,6	Y_1	Y_1座標	0~399
7,8	X_2	X_2座標	0~639
9,10	Y_2	Y_2座標	0~399
11	N	変更カラーコード数	1~8
10+2・n	CLn	旧(変更前)カラーコード	0~7
11+2・n	cln	新(変更後)カラーコード	0~7

GET BLOCK 1

対角線座標(X_1, Y_1)，(X_2, Y_2)で示される枠内の中のドットのうち，指定された色(カラーコード)のドットを1，その他の色(カラーコード)のドットを0として画面のドット情報を読み込みます．

コマンド＆パラメータ

相対値	記号	内　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$1B
3,4	X_1	X_1座標	0~639
5,6	Y_1	Y_1座標	0~399
7,8	X_2	X_2座標	0~639
9,10	Y_2	Y_2座標	0~399
11	N	指定カラーコード数	1~8
11+n	CLn	カラーコード	0~7

復帰情報

相対値	記号	内　容	値
0	E	エラーコード	〔E〕≠0のときにエラーあり
2	CN	継続フラグビット(MSB)	〔CN〕=1のときにデータが継続することを示す．
3	—		$00
3	B	今回転送バイト数	1~124
4~127	Pattern	ドット・パターン・データ	ドット・パターン

PUT BLOCK 1

対角線座標(X_1, Y_1)，(X_2, Y_2)で示される枠内に，指定された色(カラーコード)で，ドット・パターンを表示します．

コマンド＆パラメータ

相対値	記号	内　容	値
0	—		Don't Care
1	CN	継続フラグ・ビット(MSB)	$1C
2	C	コマンド・コード	〔CN〕=1のときに，データが継続することを示す．
3,4	X_1	X_1座標	0~639
5,6	Y_1	Y_1座標	0~399
7,8	X_2	X_2座標	0~639
9,10	Y_2	Y_2座標	0~399
11	CL	カラーコード	0~7
12	F	ファンクション・コード	0:PSET 1:PRESET 2:AND 3:OR 4:XOR 5:NOT
13	B	今回転送バイト数	1~114
14~127	Pattern	ドット・パターン・データ	ドット・パターン

GET BLOCK 2

対角線座標(X_1, Y_1)，(X_2, Y_2)で示される枠内より，BANK

0，BANK 1，BANK 2の画面データを読み取ります．

コマンド＆パラメータ

相対値	記号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$1D
3,4	X_1	X_1座標	0～639
5,6	Y_1	Y_1座標	0～399
7,8	X_2	X_2座標	0～639
9,10	Y_2	Y_2座標	0～399

復帰情報

相対値	記号	内　　容	値
0	E	エラーコード	〔E〕≠0のときエラーあり
1	CN	継続フラグ・ビット(MSB)	〔CN〕＝1のときに，データが継続することを示す
2	—		$00
3	B	今回転送バイト数	1～124
4～127	Pattern	ドット・パターン・データ	ドット・パターン

エラー内容については，エラーコード表を参照．

PUT BLOCK 2

対角線座標(X_1, Y_1)，(X_2, Y_2)で示される枠内のそれぞれのバンクに，ドット・パターンを表示します．

コマンド＆パラメータ

相対値	記号	内　　容	値
0	—		Don't Care
1	CN	継続フラグ・ビット(MSB)	〔CN〕＝1のときにデータが継続することを示す
2	C	コマンド・コード	$1E
3,4	X_1	X_1座標	0～639
5,6	Y_1	Y_1座標	0～399
7,8	X_2	X_2座標	0～639
9,10	Y_2	Y_2座標	0～399
11	—		Don't Care
12	F	ファンクション・コード	0：PSET　2：AND 3：OR　4：XOR 5：NOT
13	B	今回転送バイト数	1～114
14～127	Pattern	ドット・パターン・データ	ドット・パターン

GRAPHIC CURSOR

オペレータがグラフィック・カーソル(＋印)を制御して指定した，画面上の座標値を読み取ります．

コマンド＆パラメータ

相対値	記号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$1F
3	CL	カラーコード	0～7
4	N	要求座標数	1～10
5,6	X	X座標	0～639
7,8	Y	Y座標	0～399

復帰情報

相 対 値	記号	内　容	値
0	E	エラーコード	〔E〕≠0のときエラーあり
1,2	—		不定，$00
3+4•(n−1) 4+4•(n−1)	Xn	Xn座標	0～639
5+4•(n−1) 6+4•(n−1)	Yn	Yn座標	0～399

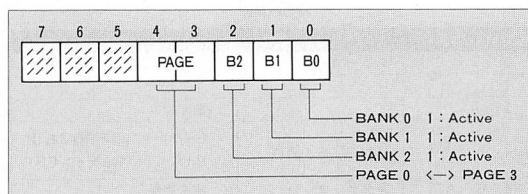
エラー内容については，エラーコード表を参照．

SELECT ACTIVE VRAM

グラフィック・コマンドの処理の対象となるVRAMを指定します．このコマンドにて処理の対象に指定されていないVRAMのページおよびバンクは，グラフィック・コマンドのカラーコード，ファンクション・コードにかかわらず読み書きされません．

コマンド＆パラメータ

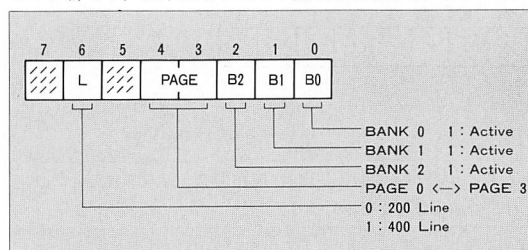
相対値	記号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$21
3	SF	選択フラグ	

選択フラグ(SF)は，アクティブVRAMを指定するためのフラグ群であり，8ビットのうちの下位5ビットが有効です，以下にそのビット構成を示します．

SELECT DISPLAY VRAM

画面に表示されるVRAMを指定します．

コマンド＆パラメータ

相対値	記号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$22
3	SF	選択フラグ	

選択フラグ(SF)は，ディスプレイVRAMを指定するためのフラグ群です．以下にそのビット構成を示します．

GET BLOCK 3

対角線座標 (X_1, Y_1)，(X_2, Y_2)で示される枠内の指定された色(カラーコード)のドットと，コンソール画面の指定された色(カラーコード)の文字を構成するドットの両方を同時に読み取ります．

コマンド＆パラメータ

相対値	記号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コール	$23
3,4	X_1	X_1座標	0～639
5,6	Y_1	Y_1座標	0～399
7,8	X_2	X_2座標	0～639
9,10	Y_2	Y_2座標	0～399
11	N	グラフィック・カラーコード数	1～8
12～19	CLn	グラフィック・カラーコード	0～7
20	NL	テキスト・カラーコード数	1～8
21～28	CTn	テキスト・カラーコード	0～7

復帰情報

相対値	記号	内　　容	値
0	E	エラーコード	〔E〕≠0のときエラーあり
1	CN	継続フラグ・ビット(MSB)	〔CN〕＝1のときにデータが継続することを示す
2	—		$00
3	B	今回転送バイト数	1～124
4	Pattern	ドット・パターン・データ	ドット・パターン

エラー内容については，エラーコード表を参照．

READ DISPLAY STATUS

グラフィックVRAMの動作状態を読み取ります.

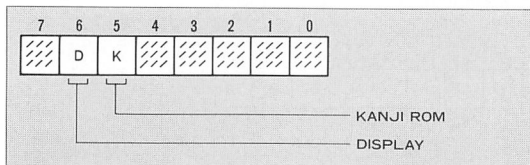
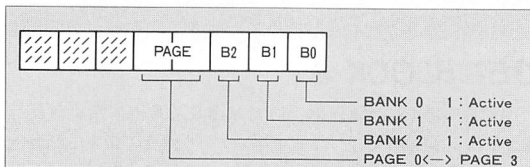
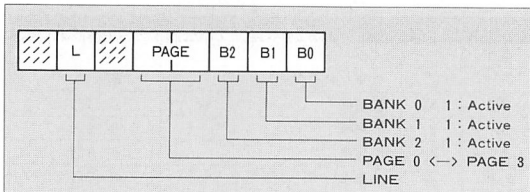
コマンド&パラメータ

相対値	記号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$24

復帰情報

相対値	記 号	内　　容	値
0	E	エラーコード	〔E〕≠0のときエラーあり
1	—		不　定
2	—		$00
3	HD	ハードウェア状況	Bit 5 ＝ 1　漢字ROMあり Bit 6 ＝ 1　400ラインCRT
4	ACT	アクティブ・ページ	本文参照
5	DIS	ディスプレイ・ページ	本文参照
6	—		不　定
7	BC	背景色カラーコード	0～7
8,9	PAGE 0	スタート・アドレス	$00～$8000
10,11	PAGE 1	スタート・アドレス	$00～$8000
12,13	PAGE 2	スタート・アドレス	$00～$8000
14,15	PAGE 3	スタート・アドレス	$00～$8000

エラー内容についてはエラーコード表を参照.

ハードウェア状況(HD)は, ハードウェアの接続状況を示すもので, 漢字ROMの有無, CRTディスプレイのライン数を示します.

アクティブ・ページ(ACT)は, 現在アクティブVRAMに設定されているVRAMを示します.

ディスプレイページ(DIS)は, 現在ディスプレイVRAMに設定されているVRAMを示します.

ERASE GRAPHIC VRAM

グラフィックVRAMのすべてのドットを背景色カラーコードに設定します.

コマンド&パラメータ

相対値	記号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$25
3	BC	背景色カラーコード	0～7

SET GRAPHIC START ADDRESS

現在アクティブVRAMに設定されているページのスタート・アドレスを設定します.

コマンド&パラメータ

相対値	記号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$26
3	SF	設定条件	0：スタート・アドレスの設定 1：スタート・アドレスの増減
4	ADDR	スタート・アドレス・データ	16ビット・データ

KANJI SYMBOL

漢字を指定座標より, 指した角度, 大きさ(倍率)で表示します. 漢字を表示するときには, 文字を構成するドット以外は変化しません.

コマンド&パラメータ

相対値	記 号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$27
3	CL	カラーコード	0～7
4	F	ファンクション・コード	0：PSET　1：PRESET 2：AND　3：OR 4：XOR　5：NOT
5	A	アングル・コード	0：通常(ノーマル) 1：90°左回転 2：180°左回転 3：270°左回転
6	W	文字横幅倍率	1のとき16ドット表示
7	H	文字高倍率	1のとき16ドット表示
8,9	X	X座標	0～639
10,11	Y	Y座標	0～399
12	N	漢字文字数	1～40
11+2･n 12+2･n	CODEn	漢字文字コード	JISコード

READ FONT

漢字またはASCIIコードで示されるコンソール表示文字の文字フォントを読み取ります.

コマンド&パラメータ

相対値	記号	内　　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$28
3	SF	選択フラグ	$00 ：キャラクタ(8×8) $01 ：キャラクタ(8×16) $FF：漢字(16×16)
4,5	CD	漢字JISコードまたはASCIIコード	8または16ビット・コード・データ

選択フラグ(SF)は, 読み取るフォントの選択をします. SFの値が$00のときには8×8ドットのコンソール表示文字のフォントを, $01のときには8×16ドットのコンソール表示文字のフォントを, $FFのときには漢字のフォントを読み取ります.

漢字JISコードまたはASCIIコード(CD)は, 漢字およびコンソール表示文字のコードです. 漢字の場合には, 16ビットのJISコードが, コンソール表示文字の場合には, 8ビットのASCIIコードが用いられます. ASCIIコードの場合には, CDの上位8ビット(相対値4のバイト)に設定します.

復帰情報

相 対 値	記 号	内　　容	値
0	E	エラーコード	〔E〕≠0のときエラーあり
1	—		不　定
2	—		$00
3～34	Pattern	フォント・データ	ドット・パターン

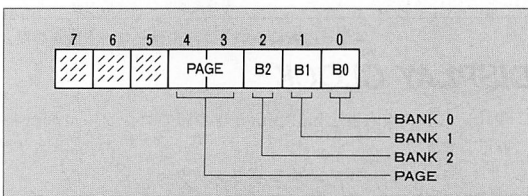
エラー内容については, エラーコード表を参照.

COPY GRAPHIC VRAM

グラフィックVRAMの，指定した部分の内容を，別の部分に複写します．

コマンド＆パラメータ

相対値	記 号	内　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$33
3	SC	ソース・ページ	
4,5	SCADR	ソース・アドレス	$0000～$3FFF
6,7	BYTES	1行のバイト数	1～80
8,9	LINES	複写行数	1～200
10	DS	ディスティネーション・ページ	
11,12	DSADR	デスティネーション・アドレス	$0000～$3FFF

ソース・ページ（SC）は，複写元となるVRAMのページおよびバンクを指定します．同様に，ディストネーション・ページ（DS）は，複写先のVRAMのページおよびバンクを指定します．

KANJI OUT

コンソール画面の現在のカーソル位置に漢字を表示します．

コマンド＆パラメータ

相対値	記 号	内　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$34
3,4	CODE	漢字文字コード	JISコード

DEFINE EXTERNAL FONT OF KANJI

漢字の外字フォント（JISコードに定義されていない文字のフォント）を定義します．

コマンド＆パラメータ

相対値	記 号	内　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$35
3,4	CODE	外字コード	外字コード
5～36	FONT	文字フォント	ドット・パターン

外字コード（CODE）は，登録する文字のコードであり，JISコードに定めてある85区～94区までの940文字分のコードを使用します．

LINE 2

LINEと同じ様に，対角点（X_1, Y_1），（X_2, Y_2）を結ぶ直線または，（X_1, Y_1），（X_2, Y_2）を対角線とする四角形を描きますが，LINE 2は，LINEとは直線補間のアルゴリズムが異なり，直線を引くときに少しでもその線にかかる点すべてを表示します．

コマンド＆パラメータ

相対値	記号	内　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$36
3	CL	カラーコード	0～7
4	F	ファンクション・コード	0：PSET　1：PRESET　2：AND　3：OR　4：XOR
5,6	X_1	X_1座標	0～639
7,8	Y_1	Y_1座標	0～399
9,10	X_2	X_2座標	0～639
11,12	Y_2	Y_2座標	0～399
13	B	ボックス・フラグ	0：直線 1：四角形 2：四角形ぬり

キーボードコマンド詳細

INKEY 1

キーコード・バッファより8ビット・キーコードを読み込みます．

コマンド＆パラメータ

相対値	記号	内　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$29
3	CF	制御フラグ	0～5

キーコード・バッファより8ビットのキーコードを読み取ります．キーコード読み取りのときに，エディット・キーのコード変換およびPFキー生成文字列への交換が行なわれます．

制御フラグ(CF)の値によって，次の動作が行なわれます．

0：キーコード・バッファからキーコードを読み込みます．
1：キーコード・バッファからキーコードを読み込みます．キーコード・バッファが空の場合にキー入力を待ちます．
2：キーコード・バッファを空にします．
3：キーコード・バッファを空にした後に，キー入力を待ちます．　　(以上形式1)
4：現在のキーコード・バッファのINKEY 1で読み込めるキーコードの数を求めます(形式2)．
5：現在のキーコード・バッファのINKEY 2で読み込めるキーコードの数を求めます(形式2)．

復帰情報

<形式1>

形式値	記号	内　容	値
0	E	エラーコード	〔E〕≠0のときエラーあり
1	—		不　定
2	—		$00
3	K	キーコード(8ビット)	キーコード
4	EN	キーコードの有無	0：なし　1：あり

エラー内容については，エラーコード表を参照．

<形式2>

相対値	記号	内　容	値
0	E	エラーコード	〔E〕≠0のときエラーあり
1	—		不　定
2	—		$00
3,4	FUN	キーコード数	キーコードの数

エラー内容については，エラーコード表を参照．

DEFINE PF STRING

プログラマブル・ファンクション（PF）キーに，PFキー生成文字列用の文字列を定義します．

コマンド＆パラメータ

相対値	記 号	内　容	値
0,1	—		Don't Care
2	C	コマンド・コード	$2A
3	NO	PFキー番号	1～10
4	N	文字数	0～15
5～19	String	文字列	文字列

GET PF STRING

プログラマブル・ファンクション（PF）キーに定義されている，PFキー生成文字列を読み取ります．

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$2B
3	NO	PFキー番号	1～10

復帰情報

相対値	記号	内容	値
0	E	エラーコード	〔E〕≠0のときエラーあり
1	—		不定
2	—		$00
3	NO	PFキー番号	1～10
4	N	文字数	0～15
5～19	String	文字列	文字列

エラー内容については，エラーコード表を参照．

SET PF INTERRUPTION

PFキー割り込み機能を用いるPFキーを設定します．

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$2C
3,4	IC	割り込み制御フラグ	10ビット・データ

INKEY 2

キーコード・バッファより9ビット・キーコードを読み込みます．

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$2D

復帰情報

相対値	記号	内容	値
0	E	エラーコード	〔E〕≠0のときエラーあり
1	—		不定
2	—		$00
3,4	KEYCODE	キーコード	9ビット・キーコード

ALLOCATE KEY BOARD

キーボードが，Display Sub Systemに配置されているか，メインCPU側に配置されているかをDisplay Sub Systemに通知します．

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$2E
3	ALF	アロケーション・フラグ	〔ALF〕=0 Display Sub System 〔ALF〕≠0 メインCPU

タイマ機能解説

SET TIMER

タイマの各レジスタに値を設定します．

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$3D
3	RS	レジスタ選択フラグ	
4	TC	制御レジスタ	制御レジスタの設定値
5～8	T1	時計レジスタ	時計レジスタの設定値
9～12	T1-I	割り込み予約時刻レジスタ	割り込み予約時刻レジスタの設定値
13～16	T2	インターバル・カウンタ	インターバル・カウンタの設定値
17～20	T2-D	再設定値レジスタ	再設定値レジスタの設定値

READ TIMER

タイマの各レジスタの値を読み取ります．

復帰情報

相対値	記号	内容	値
0	E	エラーコード	〔E〕≠0のときエラーあり
2	—		$00
1,3	—		不定
4	TC	制御レジスタ	制御レジスタの内容
5～8	T1	時計レジスタ	時計レジスタの内容
9～12	T1-I	割り込み予約時刻レジスタ	割り込み予約時刻レジスタの内容
13～16	T2	インターバル・カウンタ	インタバル・カウンタの内容
17～20	T2-D	再設定値レジスタ	再設定値レジスタの内容

エラー内容については，エラーコード表を参照．

DISPLAY CLOCK

時計レジスタの内容をhh：mm：ssの形でコンソール画面上に表示します．

コマンド＆パラメータ

相対値	記号	内容	値
0,1	—		Don't Care
2	C	コマンド・コード	$40
3	F	表示フラグ	〔F〕=0のときに非表示 〔F〕≠0のときに表示
4	CL	表示カラーコード	0～7
5	X	X座標	0～79(コンソール座標)
6	Y	Y座標	0～24(コンソール座標)

CONTINUE（継続コマンド）

Display Subsystem　出力の場合

転送するデータのバイト数が多く，1回の転送ですべてのデータを転送できない場合などに，残りのデータを継続して転送するのに用います．

コマンド＆パラメータ

相対値	記号	内容	値
0	—		Don't Care
1	CN	継続フラグ・ビット(MSB)	〔CN〕=1のときにデータが継続することを示す．
2	C	コマンド・コード	$64
3	B	今回転送バイト数	1～124
4～127	DATA	転送データ	データ

エラーコード表

エラーコード	エラー内容
$3C	アボート発生
$3D	コンソール座標値の誤り
$3E	作業領域の不足
$3F	グラフィック座標値の誤り
$40	ファンクション・コードの誤り
$41	座標値の誤り
$42	文字数の誤り
$43	色数の誤り
$44	PFキーの番号の誤り
$45	コマンド・パラメータの誤り
$46	コマンド・コードの誤り キーボードがメインCPUに配置されている

FM-11 ハード仕様

メモリ・マップとメモリ管理方式

メイン・ボード　メモリ・マップ

メイン・ボードはメイン・ブロックとサブブロックからなり，メイン・ブロックは128Kバイトのメイン・メモリ，I/Oインターフェイスの一部および拡張スロットから成り，サブブロックはサブCPUを含むディスプレイ・サブシステムを構成します．

図1　メイン・ブロック　メモリ・マップ

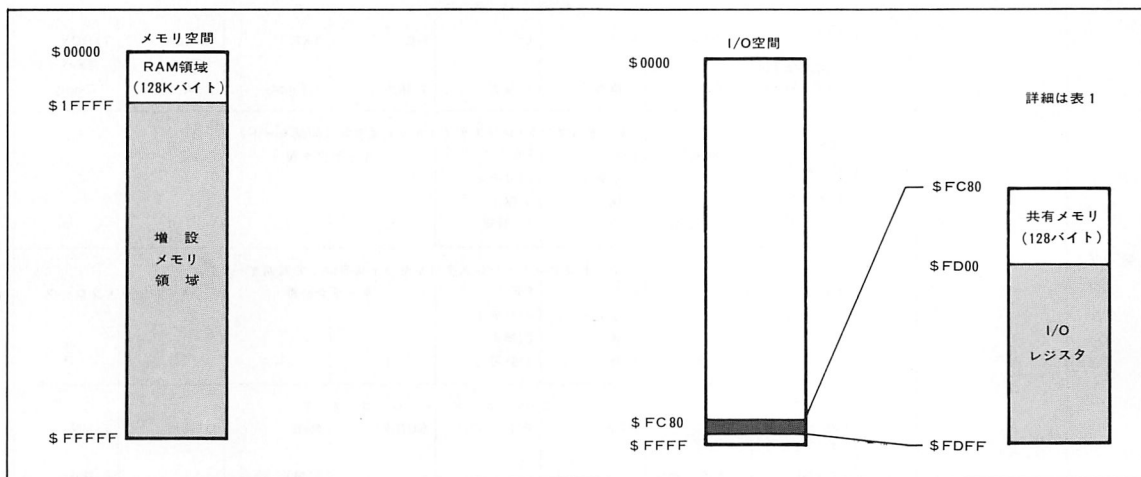



表1　メイン・ブロック　I/Oマップ

アドレス	内　容	R/W	ビット構成 bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$FD00	キーボード・データ	R	あ　き							D8
	オーディオカセット，ラインプリンタコントロール・レジスタ	W	LP SLCTIN 0：セレクト 1：非セレクト	LPSTRB データを送るためのストローブ・パルス ⎍	未使用	あ	き	未使用	リモート 0：OFF 1：ON	カセットライト・データ
$FD01	キーボード・データ	R	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
	ラインプリンタ・データ	W	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$FD02	オーディオカセット，ラインプリンタフラグ・レジスタ	R	カセットリード・データ	LPセレクト 1：セレクト 0：非セレクト	Busy dipsw 1：Busy2 0：Busy	未使用	PE 0：紙あり 1：紙なし	*KACKIRQ 0：あり 1：なし	LPERROR 1：ERROR 0：Not ERROR	LP Busy 0：ready 1：busy
	割り込み許可レジスタ（リセット時0）	W	キー割り込み 0：サブ 1：メイン	拡張メモリパリティ 0：禁止 1：許可	5"FDD 0：禁止 1：許可	タイマIRQ 0：禁止 1：許可	RS-232C SYNDET 0：禁止 1：許可	RxRDY 0：禁止 1：許可	0：禁止 1：許可	LPACK 0：禁止 1：許可
$FD03	IRQフラグレジスタ	R	RS・232C 0：割り込みなし 1：割り込みあり	キーボード 0：割り込みなし 1：割り込みあり	拡張IRQ 0：割り込みなし 1：割り込みあり	09DMA 0：割り込みなし 1：割り込みあり	5"FD 0：割り込みなし 1：割り込みあり	未使用	LPACK 0：割り込みなし 1：割り込みあり	PTM 0：割り込みなし 1：割り込みあり
	ブザーレジスタ（リセット時0）	W	ブザー 0：OFF 1：ON	あ　き						
$FD04	FIRQフラグレジスタ	R	外部FIRQ 0：割り込みなし	あ　き				未使用	Breckキー 0：キーOFF	ATTENT 0：割り込みなし

表1

アドレス	内　容	R／W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
			ビット構成							
$FD04			1：割り込み あり	あき				未使用	1：キーON	1：割り込み あり
	メイン・システム アテンションアック レジスタ	W	このアドレスにライトすることで，サブのメイン・アックビットがONになり，メインにかかっていた割り込みが解除される．							
$FD05	インターフェイス レジスタ	R	サブBusy 1：Busy 0：Ready	LP2検出 0：あり 1：なし	LP1検出 0：あり 1：なし	IEEE-488 検出 0：あり 1：なし	RS-232C検出 COM3,4 0：あり 1：なし	COM1,2 0：あり 1：なし	8"FDA検出 0：あり 1：なし	HD検出 0：あり 1：なし
	サブZ80 インターフェイス レジスタ	W	サブホールト 1：要求 0：解除	キャンセル IRQストロー ブ・パルス⎍	あき				未使用	Z80 0：メイン 1：Z80
$FD06	RS-232C (COM0：) インターフェイス	R	シリアル受信データ D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
		W	シリアル送信データ D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$FD07		R	ステータス・レジスタ DSR 0:DSR端子H 1:DSR端子L	SYNDET 1:検出	FE 1:検出	OE 1:検出	PE 1:検出	TXE 1:Empty	RxRDY 1：RxRDY	TxRDY 1:TXバッファ Empty
		W	モードコマンド・レジスタ（リセット後有効：同期モード） SCS SYNC char 0:ダブル 1:シングル	SYNDET 1:SYNC検出	EP パリティ 0:偶 1:奇	PE パリティ 0:禁止 1：許可	キャラクタ長 L_2	L_1	0	0
			モードコマンド・レジスタ（リセット後有効：非同期モード） ストップ・ビット S_2	S_1	EP パリティ 0:偶 1:奇	PE パリティ 0:禁止 1:許可	キャラクタ長 L_2	L_1	ボーレート・クロック B_2	B_1
			コマンド・レジスタ EH 1:SYNC サーチ	IR 1:内部リセット	RTS 1:$\overline{RTS}$=0	ER 1:リセット	SBRK 1：TxD=L	RxE 0:禁止 1:許可	DTR 1:$\overline{DTR}$=0	TxEN 0:禁止 1:許可
$FD08	※ 音声合成 インターフェイス	W	あき		カウンタ・ クリア 1:クリア	あき	VReset 1:DSP リセット	INT 1:有効	STOP 1:ストップ	START 1:スタート
$FD09		W	CM3 コマンド・データ	CM2	CM1	CM0	DATA3 データ	DATA2	DATA1	DATA0
$FD0A		R	$\overline{Busy}$ 0:busy 1:ready	あき						
$FD0B		W	音声データ　メモリ・バッファ D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$FD0C	※ ADコンバータ	R	あき							変換データ Dout
		W	あき				$\overline{CS}$ チップ・セレクト 0:セレクト	RS 0:0～5/8V 1:0～5/2V	チャネル・セレクト C_1	C_0
$FD0D		R								
		W								
$FD0E		R								
		W								
$FD0F		R								
		W								

表 1

アドレス	内　容	R/W	ビット構成							
			bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$FD10	※ バブルメモリ・ インターフェイス									
$FD12										
$FD12										
$FD13										
$FD14										
$FD15										
$FD16										
$FD17										
$FD18	ミニフロッピー インターフェイス (EX, ADのみ)	R	ステータス・レジスタ (type I)							
			Not Ready .1:NotReady	Write Protect 1:Protect	Head Load 1:ON	Seekエラー 1:エラー	CRCエラー 1:エラー	Track00 1:00	Index 1:ON	Busy 1:busy
			ステータス・レジスタ (type II, III)							
			Not Ready 1:Not Ready	Write .Protect 1:Protect	レコード・タイプ 0:データ・マーク 1:デリーテッドアドレス・マーク	Record Not found 1:Not found	CRCエラー 1:エラー	ロスト・データ 1:ロスト	データ・リクエスト 1:セット	Busy 1:busy
		W	コマンド・レジスタ							
			type I Restore　0 0 0 0 1 V 0 0							
			Seek　0 0 0 1 1 V 0 0							
			Step　0 0 1 U 1 V 0 0							
			Stepin　0 1 0 U 1 V 0 0							
			Stepout　0 1 1 U 1 V 0 0							
			type II Read　1 0 0 m × E 0 0							
			Write (データ・マーク)　1 0 1 m × E 0 0							
			Write (デリーテッドデータ・マーク)　1 0 1 m × E 0 1							
			type III Read address　1 1 0 0 0 1 0 0							
			Write track　1 1 1 1 0 1 0 0							
			type IV Force interrupt　1 1 0 1 I_3 I_2 I_1 I_0							
			V:veryfy. U:update. m:multiple record E:30mS delay, I:interrupt							
$FD19		R/W	トラック・レジスタ							
$FD1A		R/W	セクタ・レジスタ							
$FD1B		R/W	データ・レジスタ							
$FD1C		R	DRQ 0:OFF 1:ON	IRQ 0:OFF 1:ON	あき					SIDE 0 1
		W	あき							SIDE 0 1
$FD1D		R/W	モータ 0:OFF 1:ON	ドライブ・ディスエーブル 0:イネーブル 1:ディスエーブル	あき				ドライブ No 0〜3	
		リセット時	0	1					0	0
$FD1E			あき							
$FD1F		R	DRQ 0:OFF 1:ON	IRQ 0:OFF 1:ON	あき					SIDE 0 1
$FD18 〜 $FD1E	BASIC ROM インターフェイス (STのみ)		あき							
$FD1F		R	データ・レジスタ							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
		W リセット時 0	ROMセレクトレジスタ							
			あき						ROM 2 1:セレクト 0:非セレクト	ROMI 1:セレクト 0:非セレクト

表 1

アドレス	内　容	R/W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
			ビット構成							
$FD20	アテンション データ・レジスタ	R	アテンション・データ #0							
$FD21		R	アテンション・データ #1							
$FD22		R	アテンション・データ #2							
$FD23	ソフトウェア読み出し用ディップスイッチ・レジスタ	R	デイップスイッチ・ステータス 0:ON 1:OFF SW8	SW7	SW6	SW5	SW4	SW3	SW2	SW1
	LPAck クリア・レジスタ	W	あき					LPAckクリア・レジスタ 1:クリア 0:何もせず LP2	LP1	LP0
$FD24	※ RS-232C (COM1:)	R	受信データ							
		W	送信データ							
$FD25		R	ステータス							
		W	コマンド							
$FD26	※ RS-232C (COM2:)	R	受信データ							
		W	送信データ							
$FD27		R	ステータス							
		W	コマンド							
$FD28	※ RS-232C (COM3:)	R	受信データ							
		W	送信データ							
$FD29		R	ステータス							
		W	コマンド							
$FD2A	※ RS-232C (COM4:)	R	受信データ							
		W	送信データ							
$FD2B		R	ステータス							
		W	コマンド							
$FD2C	※拡張IRQ フラグ・レジスタ	R	拡張RS 0:割り込みあり 1:割り込みなし	8"FD 0:割り込みあり 1:割り込みなし	LP1Ack 0:割り込みあり 1:割り込みなし	LP2Ack 0:割り込みあり 1:割り込みなし	IEEE-488 0:割り込みあり 1:割り込みなし	HD 0:割り込みあり 1:割り込みなし	reserve	
	※拡張IRQ 許可レジスタ (リセット時0)	W	RS-232C COM1～4 SYNDET 0:禁止 1:許可	RxRDY 0:禁止 1:許可	TxRDY 0:禁止 1:許可	LP1, 2 Ack 0:禁止 1:許可	8" FDC 0:禁止 1:許可	あき		
$FD2C										
$FD2E										
$FD2F	※ Z80バンク・レジスタ (リセット時0)	W	あき				Z80バンク・レジスタ A_{19}	A_{18}	A_{17}	A_{16}
$FD30	※ 標準フロッピィー インターフェイス	R	ステータス・レジスタ (type I)							
			Not Ready 1:Not Ready	Write Protect 1: Protect	Head Load 1:ON	Seekエラー 1:エラー	CRCエラー 1:エラー	Track00 1:00	Index 1:ON	Busy 1: busy
			ステータス・レジスタ (type II, III)							
			Not Ready 1: Not Ready	Write Protect 1: Protect	レコードタイプ 0:データ・マーク 1:デリーテッドアドレス・マーク	Record Not found 1: Not found	CRCエラー 1:エラー	ロスト・データ 1:ロスト	データ・リクエスト 1:セット	Busy 1: busy

表1

アドレス	内　　容	R/W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
			ビット構成							
$FD30		W	コマンド・レジスタ type Ⅰ Restore　0 0 0 0 1 V 0 0 Seek　0 0 0 1 1 V 0 0 Step　0 0 1 U 1 V 0 0 Stepin　0 1 0 U 1 V 0 0 Stepout　0 1 1 U 1 V 0 0 type Ⅱ Read　1 0 0 m × E 0 0 Write（データ・マーク）　1 0 1 m × E 0 0 Write（デリーテッド・データ・マーク）　1 0 1 m × E 0 1 type Ⅲ Read address　1 1 0 0 0 1 0 0 Write track　1 1 1 1 0 1 0 0 type Ⅳ Force interrupt　1 1 0 1 I_3 I_2 I_1 I_0 V:veriy. U:update. m:multiple record. E:30mS delay. I:interrupt							
$FD31	※標準フロッピィーインターフェイス	R/W	トラック・レジスタ							
$FD32		R/W	セクタ・レジスタ							
$FD33		R/W	データ・レジスタ							
$FD34		R/W	あ　き							SIDE 0 1
$FD35		R/W	ドライブ・レジスタ ドライブ・セレクト，ディスエーブル・フラグ	あ				き	ドライブNo.	
$FD36										
$FD37		R	DRQ 1:ON	IRQ 1:ON	あ			き		
$FD38	Programable Timer	R	ノーオペレーション							
		W	コントロール・レジスタ2のビット0＝0のとき　コントロール・レジスタ3に書き込み							
			コントロール・レジスタ2のビット0＝1のとき　コントロール・レジスタ1への書き込み							
$FD39		R	ステータス・レジスタ 複合割り込みフラグ					タイマ3 割り込み	タイマ2 割り込み	タイマ1 割り込み
		W	コントロール・レジスタ 出力許可 0:禁止 1:許可	割り込み許可 1:許可	動作モード			カウント・モード 0:16ビット 1: 8ビット×2	クロック源 0:C 1:E	0:CR3 1:CR1
$FD3A $FD3B		R/W	タイマ1カウンタ　アクセスは16ビット単位で行なう							
$FD3C $FD3C		R/W	タイマ2カウンタ　アクセスは16ビット単位で行なう							
$FD3E $FD3F		R/W	タイマ3カウンタ　アクセスは16ビット単位で行なう							
$FD98	パレット・レジスタ（注2）	R/W	あ		き		I	Color ♯0 G	R	B
$FD99			あ		き		I	Color ♯1 G	R	B
$FD9A			あ		き		I	Color ♯2 G	R	B
$FD9B			あ		き		I	Color ♯3 G	R	B
$FD9C			あ		き		I	Color ♯4 G	R	B

表 1

アドレス	内　容	R/W	ビット構成 bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$FD9D				あ	き		Color #5 I	G	R	B
$FD9E				あ	き		Color #6 I	G	R	B
$FD9F				あ	き		Color #7 I	G	R	B
$FDA0	ビデオ出力コントロール・レジスタ(リセット時0)	W		あ	き		グラフィックを Greenへ 0:出力しない 1:出力する	グラフィックを カラーへ 0:出力しない 1:出力する	キャラクタを Greenへ 0:出力しない 1:出力する	キャラクタを カラーへ 0:出力しない 1:出力する

注1) "あき"はここにハードウェアがないこと,"未使用"はバッファ・ラッチがつながっているが使っていないことを示す. ※印はオプションである.
注2) 下位4ビットのみR/W可. ただし, リセットによって次のようにイニシャライズされる. IはIntensity.

表1-1

Color#	入力ピン IN3 or A3	IN2 or A1	IN1 or A0	リセット時の設定 I	G	R	B
0	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1
2	0	1	0	1	0	1	0
3	0	1	1	1	0	1	1
4	1	0	0	1	1	0	0
5	1	0	1	1	1	0	1
6	1	1	0	1	1	1	0
7	1	1	1	1	1	1	1

パレット・レジスタとビデオ出力コントロール・レジスタ

パレット・レジスタとビデオ出力コントロール・レジスタは, メインCPUのアドレス空間に属し, 6809CPUボード使用時はそれぞれ$FD98～9Fと$FDA0, 8088CPUボード使用時はI/Oアドレスの0FD98H～9FHと0FDA0Hになる. レジスタの構成を表1-1に示す.

6809CPUボード使用時のメモリ・マップとウインドウ・レジスタ, MMR

リセット・ベクトルはブートROM (MBM2732A) のアドレス$5FFE, $5FFFのアドレスに書き込まれていなければならない. リセット中は$FFFE, $FFFFは上記ROMアドレスに割り当てられるが, リセット解除後は6809CPUボード内のRAMに割り当てられる. ただし, このRAMに書き込まれたデータはリセット・ベクトルにはならない.

また, リセット時にMemory Mapping Register (以下MMR

図2　サブブロック　メモリ・マップ

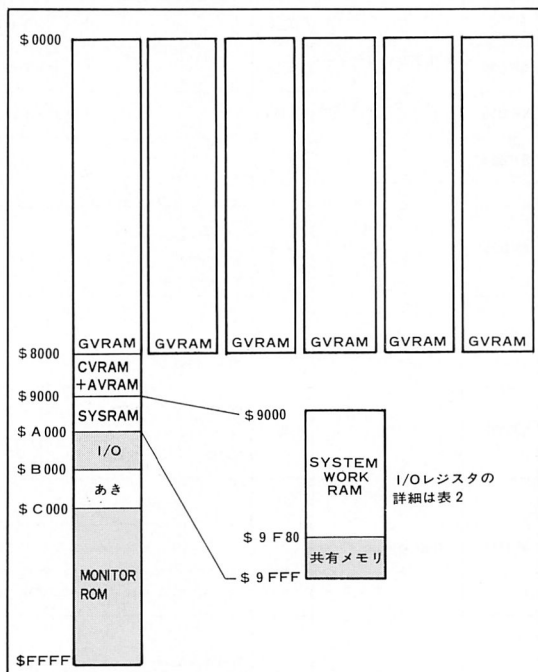



表2　サブシステムI/Oマップ

アドレス	内　容	R/W	ビット構成 bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$AFE0	MIXレジスタ(リセット時0)	R/W	あ	き	カーソル表示 0:LSB=0 1:LSB=1	ライトペン 0:レベル 1:トリガ	WIDTH 0:40 1:80	FLASH 0:OFF 1:ON	INSLED 0:OFF 1:ON	Busyflog 0:busy 1:ready
$AFE1	グラフィックVRAMアップデイト・レジスタ	R/W	リードアウト・コントロール RC 2	RC 1	バイトゼロ・ライト 0:する 1:しない RAM 3	RAM 2	RAM 1	RAMセレクト・ビット 0:非セレクト, 1:セレクト RAM 3	RAM 2	RAM 1
$AFE2	グラフィックVRAMディスプレイモード・レジスタ(リセット時0)	W	あ　き	モード設定M1 0:200ドット 1:400ドット	あ　き	ページ・セレクト PS 2	PS 1	RAMセレクト・ビット 0:非セレクト, 1:セレクト RAM 3	RAM 2	RAM 1
$AFE3	グラフィックVRAMページ・セレクト・レジスタ(リセット時0)	W		あ	き	ページ・セレクト PS 2	PS 1	あ		き
$AFE4	FIRQクリア・レジスタ (リード後0)	R	ライトペン 0:割り込みなし 1:割り込みあり		あ			き		キーボード 0:割り込みなし 1:割り込みあり
$AFE5	キャンセルIRQクリア・レジスタ	R	このレジスタをリードすることでキャンセルIRQの要因をクリアできる.							
$AFE6	ステータス表示レジスタ	R	ブランキング 0:ブランキング 1:表示中	ディスプレイ 1:400ライン 0:200ライン	漢字ROM 1:実装 0:未実装	あ	き	VSYNC 1:VSYNC中	ライトペンLSB 0:LSB=0 1:LSB=1	ライトペンSW 0:OFF 1:ON
$AFE7	アテンションFIRQフラグ・レジスタ	Don't Care	メイン・システムがAckをONにしたときにクリアされる.							

アドレス	内容	R/W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$AFE8	アクセス・スタートアドレス・レジスタ(リセット時0)	W	スタート・アドレス 上位バイト							
			あき	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
$AFE9			スタート・アドレス 下位バイト							
			A_7	A_6	A_5	A_4	A_3	A_2	A_1	あき
$AFEA	グラフィック・スタートアドレス・レジスタ(リセット時0)	W	スタート・アドレス 上位バイト							
			あき	A_{14}	A_{13}	A_{11}	A_{11}	A_{10}	A_9	A_8
$AFEB			スタート・アドレス 下位バイト							
			A_7	A_6	A_5	A_4	A_3	A_2	A_1	あき
$AFEC	Key in data レジスタ (リセット時0 リード後 0)	R	あき							D8
$FEED			キー入力データ							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$AFEE	CRTコントローラ	R/W	未使用			CRTCレジスタ・アドレス				
						D_4	D_3	D_2	D_1	D_0
$AFEF			CRTCデータ・アドレス							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$AFF0	アテンションレジスタ(リセット時Ack ON)	W	アテンション・データ ＃0							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$AFF1			アテンション・データ ＃1							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$AFF2			アテンション・データ ＃2							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$AFF3		R	メイン・アック 0:Ack OFF 1:Ack ON	未使用						
$AFF4	漢字ROM読み出しレジスタ(リセット時0)	W	あき				読み出しアドレス上位バイト			
							A_{15}	A_{14}	A_{13}	A_{12}
$AFF5			漢字ROM読み出しアドレス下位バイト							
			A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4
$AFF6		R	漢字ROM読み出しデータ上位バイト							
			D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8
$AFF7			漢字ROM読み出しデータ下位バイト							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$AFF8 $AFF9	Buzzerレジスタ(リセット時OFF)	R/W	Read でブザーON Write でブザーOFF							

図3 リセット直後の6809CPUボードメモリ・マップ

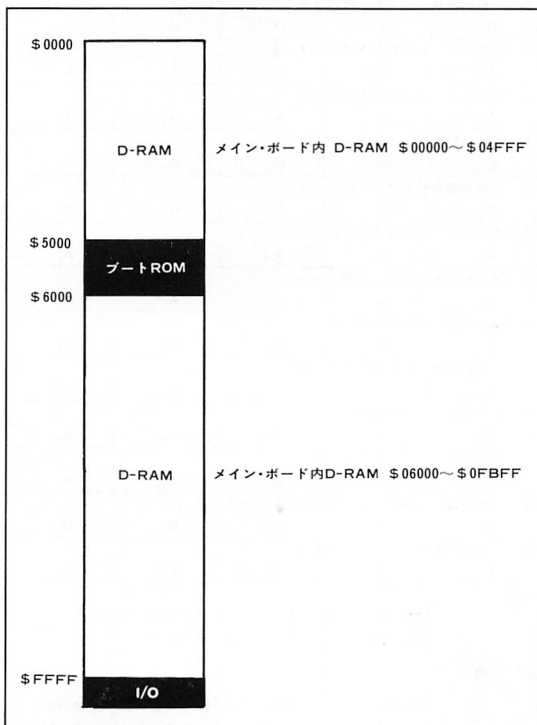



表3 MSR

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$FD93 R/W (リセット時0)							未使用	ブートROM 0:有効 1:無効

と略す)，Memory Segment Register(以下MSGと略す)，Window Offset Register (以下WORと略す) およびWindow Bank Register (以下WBRと略す) は，すべてディスエーブルになっている．

ブートROMは**表3**に示すMode Select Register (以下MSRと略す)のビット0，ブートROMビットに1を書くことで禁止し，ブートROMが占めていたエリアをメイン・ボード上のD-RAMにすることができる．このときのメモリ・マップの様子を**図4**に示す．

ウインドウ・レジスタを使ったメモリのアクセス

ウインドウ・レジスタはCPU空間の＄7C00～＄7FFFの領域と，メモリの実アドレスとの対応を取るレジスタで，WBR (Window Bank Register)，とWOR (Window Offset Register) の2つのレジスタから成る．

WBRは64Kバイト単位のどのメモリ・バンクを，ウインドウ経由でアクセスするかを指定するレジスタである．WORはWBRで指定されたBank内のどのアドレスをウインドウ領域に割り当てるかを指定するレジスタで，アドレス＄7C00に対する256バイト単位のオフセット量を書き込まなければならない．WBR，WORはリセット時に0に初期化される．このとき，ウインドウ機能は＄FD93のMSRのビット6が0になって

図4 ブートROMを禁止したときのメモリ・マップ

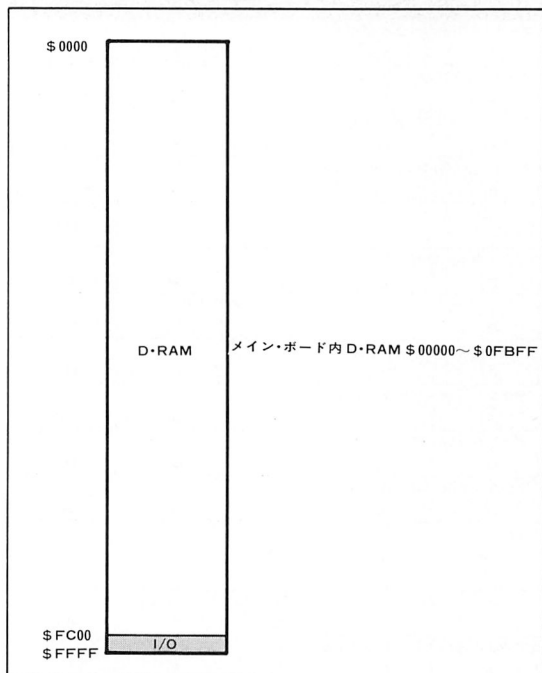



図5 6809CPUボード メモリ・マップ

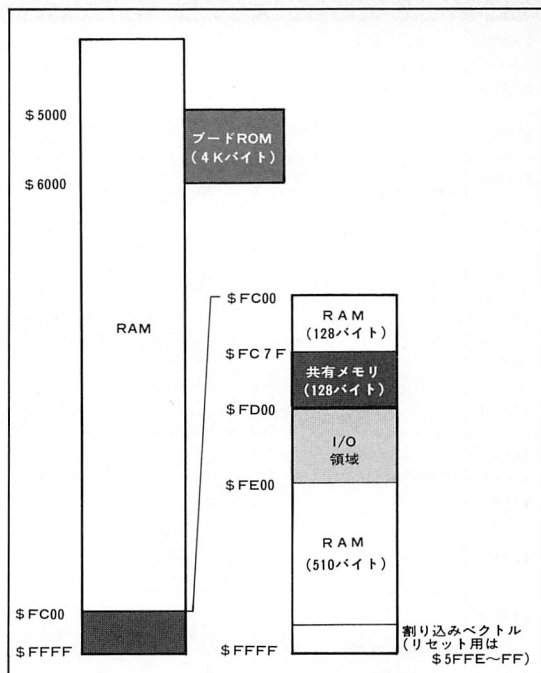



表4 6809CPUボードI/Oマップ

アドレス	内容	R/W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
			ビット構成							
$FD40	DMAC	R/W	CH0 アドレス・レジスタ上位バイト							
			A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
$FD41			CH0 アドレス・レジスタ下位バイト							
			A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
$FD42			CH0 転送語数レジスタ上位バイト							
			D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8
$FD43			CH0 転送語数レジスタ下位バイト							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$FD44			CH1 アドレス・レジスタ上位バイト							
			A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
$FD45			CH1 アドレス・レジスタ下位バイト							
			A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
$FD46			CH1 転送語数レジスタ上位バイト							
			D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8
$FD47			CH1 転送語数レジスタ下位バイト							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$FD48			CH2 アドレス・レジスタ上位バイト							
			A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
$FD49			CH2 アドレス・レジスタ下位バイト							
			A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
$FD4A			CH2 転送語数レジスタ上位バイト							
			D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8
$FD4B			CH2 転送語数レジスタ下位バイト							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$FD4C			CH3 アドレス・レジスタ上位バイト							
			A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8
$FD4D			CH3 アドレス・レジスタ下位バイト							
			A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
$FD4E			CH3 転送語数レジスタ上位バイト							
			D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8
$FD4F			CH3 転送語数レジスタ下位バイト							
			D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$FD50	DMAC	bit0～bit3はR/W	CH0. CHCR							
			DEND 1:出力中 0:未出力	Busy/Ready 1:DMA中 0:DMA完	未使用		アドレスup/down 1:down 0:up	TSC/HALT 1:TSC 0:HALT	サイクル・スチール/バースト 1:バースト 0:サイクル・スチール	R/W 1:M→IO 0:IO→M

表4

アドレス	内容	R/W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$FD51	DMAC	bit6, 7 は Rのみ	CH1. CHCR							
			DEND	Busy /Ready	未使用		アドレス up/down	TSC /HALT	サイクル・スチール /バースト	R/W
$FD52			CH2. CHCR							
			DEND	Busy /Ready	未使用		アドレス up/down	TSC /HALT	サイクル・スチール /バースト	R/W
$FD53			CH3. CHCR							
			DEND	Busy /Ready	未使用		アドレス up/down	TSC /HALT	サイクル・スチール /バースト	R/W
$FD54		R/W	PCR							
			ローテート制御 1:ローテート 0:固定	未使用			TxRQ CH3 1:イネーブル 0:ディスエーブル	TxRQ CH2 1:イネーブル 0:ディスエーブル	TxRQ CH1 1:イネーブル 0:ディスエーブル	TxRQ CH0 1:イネーブル 0:ディスエーブル
$FD55		bit0〜bit3 は R/W bit7 は Rのみ	ICR							
			IRQフラグ 1:IRQ出力0 0:IRQ出力1	未使用			TxRQ CH3 1:イネーブル 0:ディスエーブル	TxRQ CH2 1:イネーブル 0:ディスエーブル	TxRQ CH1 1:イネーブル 0:ディスエーブル	TxRQ CH0 1:イネーブル 0:ディスエーブル
$FD56		R/W	DCR							
			未例用				2/4チャネルモード 1:4ch 0:2ch	データ・チェイン・チャネル指定 00: CH0, 01:CH1 10:CH2		データ・チェイン 1:イネーブル 0:ディスエーブル
$FD57	NMI割込レジスタ	R	拡張メモリパリティ 1:なし 0:あり	DMA タイム・オーバー 0:なし 1:あり	あき					
	DMA要求コントローラ	W	リセット・コマンド							
$FD80	Memory Management Register	R/W	CPU アドレス0000〜0FFFの割り当てメモリ・アドレス							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD81			CPU アドレス1000〜1FFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD82			CPU アドレス2000〜2FFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD83			CPU アドレス3000〜3FFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD84			CPU アドレス4000〜4FFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD85			CPU アドレス5000〜5FFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD86			CPU アドレス6000〜6FFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD87			CPU アドレス7000〜7FFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD88			CPU アドレス8000〜8FFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD89			CPU アドレス9000〜9FFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD8A			CPU アドレスA000〜AFFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD8B			CPU アドレスB000〜BFFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}

表４

アドレス	内　容	R/W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
			ビット構成							
$FD8C			CPU アドレスC000～CFFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD8D			CPU アドレスD000～DFFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD8E			CPU アドレスE000～EFFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD8F			CPU アドレスF000～FBFF							
			A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}
$FD90	MMR Segment Register MM (リセット時０)	W	あき				S3	S2	S1	S0
$FD91	Window Bank Register (リセット時０)	W	あき				A_{19}	A_{18}	A_{17}	A_{16}
$FD92	Window Offset Register (リセット時０)	W	ウィンドウ・オフセット アドレス							
			OA15	OA14	OA13	OA12	OA11	OA10	OA9	OA8
$FD93	Mode Select Register (リセット時０)	R/W	MMR 0:無効 1:有効	ウィンドウ 0:無効 1:有効	あき				未使用	ブートROM 0:有効 1:無効
$FD94	DMA バンク・レジスタ	W	あき				CH0			
							A_{19}	A_{18}	A_{17}	A_{16}
$FD95			あき				CH1			
							A_{19}	A_{18}	A_{17}	A_{16}
$FD96			あき				CH2			
							A_{19}	A_{18}	A_{17}	A_{16}
$FD97			あき				CH3			
							A_{19}	A_{18}	A_{17}	A_{16}

いるため禁止されている．ウインドウ機能は，このビットを１にすることでイネーブルとなる．**表５**にMSRの構成を示す．

表５　MSR

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$FD93 R/W (リセット時０)		ウィンドウ 0:無効 1:有効						

たとえば，ウインドウ領域にメイン・メモリ・アドレス＄18800～＄18BFFを割り当てる場合は，

01→WBR，０C→WOR

をそれぞれ書き込めば良い．

また，オフセットをかけたアドレスは同一バンク内でループする．たとえば，WBR＝01，WOR＝82とした場合は，ウィンドウ領域と実メモリ・アドレスとの対応は**図６**のようになる．

メモリ・マッピング・レジスタを使ったメモリのアクセス

MMRは6809CPUの64Kバイトのアドレス空間に，１MBのメモリ空間を割り当てるもので，ダイナミック・アロケーション領域は４Kバイト単位になる．

メモリ・マッピング機能のイネーブル，ディスエーブルはMSRのbit７（**表６**）に１，０を書くことで行なわれる．

メモリ・マッピング機能は，MSG（Memory Segment Register；＄FD90）とMMR（Memory Mapping Register；

表６　MSR

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$FD93 R/W	MMR 0:無効 1:有効							

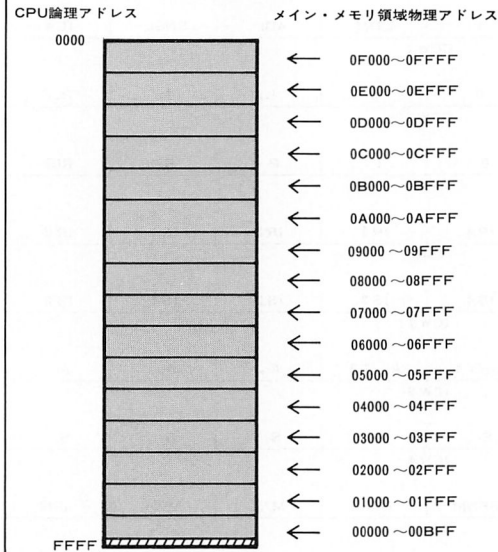
＄FD80～＄FD8F)の２種のレジスタによって行なわれている．

MMRは16バイトのレジスタの16セットから成る．どのレジスタ・セットを選択するかは，MSRの内容によって決まる．MMRの第０バイト目は，CPU空間の＄0000～＄0FFF，第１バイトは＄1000～＄1FFF，…のように順に4Kバイトごとの CPU空間に対応する．

ただし，最終バイト（＄FC00～FFFF）はアロケーションされない．MMRへの書き込み内容は，MSBから順にメイン・メモリ・アドレスの最上位ビットA19～A12になる．

MSGはリセット時にクリアされるがMMRは不定なので，MSRのMMRビットでMMRをイネーブルにする前に，ソフトウェアで初期化しなくてはならない．さらに，MMR機能はウインドウ機能との併用が可能ではあるが，＄7C00～＄7FFFはウインドウ機能が優先される．

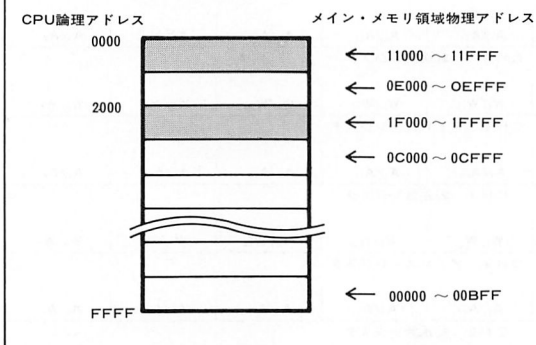
以下にMMRの使用例を示す．

❶　MSG＝00，MMRの＄FD80から順に0F，0E，0D…，02，01，00と書き込んだ場合のメモリ空間は**図７**になる．

❷　❶に続いてMMRの＄FD80に＄11を，＄FD82に＄1Fを書いた場合のメモリ空間は**図８**になる．

❸　❶に続きMSG＝01にし，MMRは❷のときに書いた通りのデータを入れておいた場合，MSG＝00，01と書き換えること

図６　同一バンク内でのループがある場合

図7　MMR使用例 ❶

図8　MMR使用例 ❷

で図7のマップと図8のマップを瞬時に切り換えることができる（図9）.

❹　例❶に，WBR＝$01，WOR＝$08とし，ウインドウ機能も合わせてイネーブルとした場合のメモリ空間は図10になる.

図10　MMRの使用例 ❹

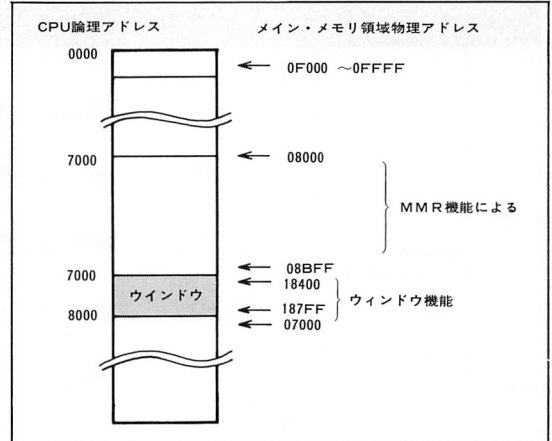
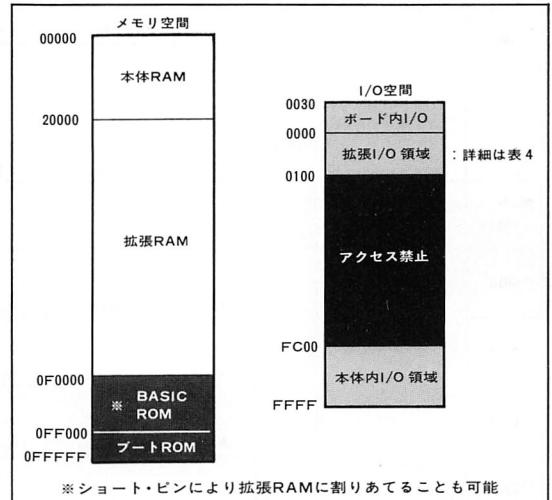



図11　8088CPUボード　メモリ・マップ

8088CPUボード使用時のメモリ・マップ（EXのみ）

8088CPUは1Mバイトのアドレス空間を生成するため，6809CPUボードの場合と異なり，アドレス生成用のハードウェア（MMR，ウインドウ）を必要としない．さらに，ブートROMの禁止も行なっていない．8088CPUボード内のI/Oはすべて8088CPUのI/Oアドレスに割りあててある．図11に8088CPUボード使用時のメモリ・マップとI/Oマップを示す．

図9　MMR使用例 ❸

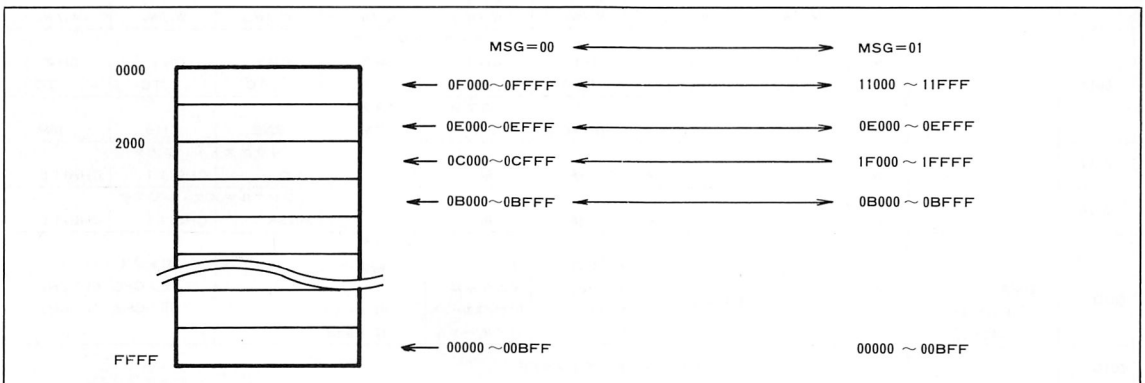



表 7　8808CPUボードI/Oマップ

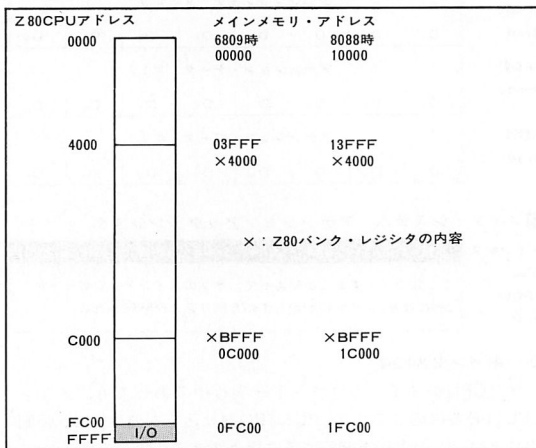
I/O アドレス	内　容	R/W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0000	割り込み コントローラ (MBL8259A)	W	ICW 1							
			A 7	A 6	A 5	1	LTIM	ADI	SNGL	IC 4
		W	OCW 2							
			R	SL	EOI	0	0	L_2	L_1	L_0
		W	OCW 3							
			X	FSMM	SMM	0	1	P	RR	RIS
		R	IRR							
			IR 7	IR 6	IR 5	IR 4	IR 3	IR 2	IR 1	IR 0
		R	ISR							
			IS 7	IS 6	IS 5	IS 4	IS 3	IS 2	IS 1	IS 0
0001		W	ICW 2							
			A_{15}/T 7	A_{14}/T 6	A_{13}/T 5	A_{12}/T 4	A_{11}/T 3	A_{10}	A_9	A_8
		W	ICW 3							
			S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0
		W	ICW 4							
			0	0	0	SFNM	BUF	M/S	AEOI	μPM
		R/W	OCW 1							
			M_7	M_6	M_5	M_4	M_3	M_2	M_1	M_0
0002～000F			ア　ク　セ　ス　禁　止							
0010	DMA コントローラ (18237A)	R/W	CH0　アドレス・レジスタ							
			A_7/A_{15}	A_6/A_{14}	A_5/A_{13}	A_4/A_{12}	A_3/A_{11}	A_2/A_{10}	A_1/A_9	A_0/A_8
0011			CH0　転送数レジスタ							
			W_7/W_{15}	W_6/W_{14}	W_5/W_{13}	W_4/W_{12}	W_3/W_{11}	W_2/W_{10}	W_1/W_9	W_0/W_8
0012			CH1　アドレス・レジスタ							
			A_7/A_{15}	A_6/A_{14}	A_5/A_{13}	A_4/A_{12}	A_3/A_{11}	A_2/A_{10}	A_1/A_9	A_0/A_8
0013			CH1　転送数レジスタ							
			W_7/W_{15}	W_6/W_{14}	W_5/W_{13}	W_4/W_{12}	W_3/W_{11}	W_2/W_{10}	W_1/W_9	W_0/W_8
0014			CH2　アドレス・レジスタ							
			A_7/A_{15}	A_6/A_{14}	A_5/A_{13}	A_4/A_{12}	A_3/A_{11}	A_2/A_{10}	A_1/A_9	A_0/A_8
0015			CH2　転送数レジスタ							
			W_7/W_{15}	W_6/W_{14}	W_5/W_{13}	W_4/W_{12}	W_3/W_{11}	W_2/W_{10}	W_1/W_9	W_0/W_8
0016			CH3　アドレス・レジスタ							
			A_7/A_{15}	A_6/A_{14}	A_5/A_{13}	A_4/A_{12}	A_3/A_{11}	A_2/A_{10}	A_1/A_9	A_0/A_8
0017			CH3　転送数レジスタ							
			W_7/W_{15}	W_6/W_{14}	W_5/W_{13}	W_4/W_{12}	W_3/W_{11}	W_2/W_{10}	W_1/W_9	W_0/W_8
0018		R	ステータ・スレジスタ							
			CH 3 REQ	CH 2 REQ	CH 1 REQ	CH 0 REQ	CH 3 TC	CH 2 TC	CH 1 TC	CH 0 TC
		W	コマンド・レジスタ							
			DACK	DREQ	WRITE	PRI	TM	ENB	CH 0	MM
0019		W	未　使　用					リクエスト・レジスタ		
								REQ	CHBIT 1	CHBIT 0
001A		W	未　使　用					シングルマスク・レジスタ		
								MASK	CHBIT 1	CHBIT 0
001B	DMA コントローラ (18237A)	W	モード・レジスタ							
			シングル・モードにすること		アドレス inc/dec 0:dec 1:inc	オート・イニシャル 0:ディスエーブル 1:イネーブル	転送 01:write 10:Read		セレクト・ビット 00：CH0, 01：CH1 10：CH2, 11：CH3	
001C		W	クリアバイト・ポインタFF							

I/O アドレス	内容	R/W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
001D		R	テンポラリ・レジスタ							
		W	マスタ・クリア							
001A		W	クリアマスク・レジスタ							
001F		W	未使用				オールマスク・レジスタ CH3	CH2	CH1	CH0
0020	DMA バンク・レジスタ	W	あき				CH0 A_{19}	A_{18}	A_{17}	A_{16}
0021			あき				CH1 A_{19}	A_{18}	A_{17}	A_{16}
0022			あき				CH2 A_{19}	A_{18}	A_{17}	A_{16}
0023			あき				CH3 A_{19}	A_{18}	A_{17}	A_{16}

表8 Z80CPUボードI/Oマップ

アドレス	内内容	R/W	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$FD2F	Z80バンク・レジスタ (リセット時0)	R/W	あき				A 19	A 18	A 17	A 16
$FD50	Z80割り込みマスク・レジスタ (リセット時0)	R/W	IRQ7 1:イネーブル 0:ディスエーブル	IRQ3,4 1:イネーブル 0:ディスエーブル	IRQ2,9 1:イネーブル 0:ディスエーブル	IRQ0 1:イネーブル 0:ディスエーブル	FIRQ0,2,3 1:イネーブル 0:ディスエーブル	IRQ5,6 1:イネーブル 0:ディスエーブル	IRQ1 FIRQ1 1:イネーブル 0:ディスエーブル	IRQ8 1:イネーブル 0:ディスエーブル

図12 Z80CPUボードのメモリ・マップ

図13 メインCPU-サブCPUインターフェイス・ブロック図

Z80CPUボード使用時のメモリ・マップ

Z80CPUも8088CPU同様I/Oアドレス空間を持つが，FM-11では使用していない．また，メモリ・アドレス空間はメインCPUが6809のときは＄00000～＄0FFFFを，8088のときは＄10000～＄1FFFFを占めるようになっている．**表8**にZ80CPUボードのI/Oマップを示す．

Z80CPUボード内のレジスタは，＄FD2FのZ80バンク・レジスタと＄FD50の割り込みマスク・レジスタのみである．

ただし，Z80バンク・レジスタだけは，メインCPUからもR/Wできるようになっている．

CPU制御

メインCPU/サブCPUのインターフェイス

メインCPU部とサブCPU部のインターフェイスは以下の3種類に分類できる．

① 共有メモリ部
② アテンション・レジスタ部
③ キャンセルIRQ部

① 共有メモリ

サブCPU側のアドレス＄9F80～＄9FFFの128バイトの領域と，メインCPUに6809を使っているときはメイン・アドレス＄FC80～＄FCFF，8088を使っているときはメインCPUのI/Oアドレスの0FC80H～0FCFFHの128バイトの領域とは，両者で共有するメモリ領域となる，たとえば，サブ側$9F80のアドレスに＄AAというデータを書き込めば，メイン・ブロック側で＄FC80(あるいは0FC80H)のアドレスをアクセスすると，同じデータである＄AAという値を読み取ることが可能である．また，メインとサブの事象が逆の関係になっても上記が成立する．

ただし，同時刻にメインとサブ側の両方からメモリをアクセスすることは不可能であるため，メイン側より共有メモリをアクセスするときに限り，まずサブCPUをホールト状態にした後に，共有メモリをアクセスしなければならないという制限をつ

ける．実際の共有メモリ領域にはCRT表示のためのライン・データ，メインからサブCPUへ通知するためのコマンド・レジスタ，サブCPUからメインCPUへ通知するためのステータス・レジスタ，タイマ・カウント・レジスタなどが含まれている．

サブCPU側でメインCPUからのコマンドを受けて命令を実行する場合に，実行が終了しない間にメインCPU側から新しいコマンドを受け付けないように，サブCPUは命令実行中はメインCPU側へBusy信号（1：命令実行中，0：実行終了Ready状態）を送出する．Busy信号はサブシステム側では＄AFE0のビット0，BUSYフラグ・ビット，メイン・システム側では＄FD05のビット7でそれぞれ見ることができる（❶，❷）．

また，メイン・システムがサブCPUにホールトをかけ，サブCPUがホールト状態になると，ハードウェアにより自動的にReady状態からぬけ，Busy信号を出す．さらに例外として，サブCPUがBusy状態であっても，メインCPUはIRQ信号によってサブCPUのBUSY状態を解除することができる．ただし，このときのBusy信号はサブCPUがソフトウェアでおろさなければならない．

❶サブシステムMIXレジスタ

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$AFEO R/W	あ	き						BUSYフラグ 0: Busy 1: Ready

❷メイン・システム　インターフェイス・レジスタ

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$FD05 Read	サブ Busy 0: Ready 1: Busy							

メインCPUとサブCPUの共有メモリ領域の同時アクセスを避けるため，メインCPUから共有メモリをアクセスする場合は，まずHALT要求信号をサブCPUへ送出し，サブCPUをホールト状態にした後にメモリ・アクセスをしなければならない．ホールト状態の解除もメインCPUが共有メモリ領域のアクセスを終了した時点でメインCPUが行なう．サブのHALT要求はメイン・アドレス＄FD05のビット7のサブホールト要求ビットに1を書くことで，また解除は0を書くことで行なう．❸にこのレジスタを示す．

❸サブ・Z80インターフェイス・レジスタ

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
メイン・システム $FD05 Write (リセット時0)	サブホールト 0:解除 1:要求	キャンセルIRQ ⎍					未使用	Z80 1:Z80 0:メイン

② アテンション・レジスタ

アテンション・レジスタによる通信は，サブシステム→メイン・システムの単方向のみ行なわれ，以下に示すような3バイト以下のデータの転送の場合のみ用いられる．

●インターバル・タイマ・クロック

●プログラマブル・ファンクション・キー

アテンション割り込みは6809CPUボードではFIRQに，8088CPUボードではINTRに，それぞれつながっている．

サブシステムはアテンション・レジスタのメインアック・ビットを見て，メイン・システムがすでにアテンション・データ・レジスタの内容を使ったことを確認したのち，アテンション・データ・レジスタにデータを書き込み，❹に示すサブアドレス＄AFE7，アテンションFIRQフラグ・レジスタをアクセスしてメイン・システムにFIRQ（あるいはINTR）をかける．このとき，ハードウェアでメインアック・ビットはOFFになる．

メイン・システムは割り込み処理ルーチン内で，❺に示すFIRQフラグ・レジスタのビット0のアテント・フラグを見てサブからのアテンション割り込みであることを確めた上で，❻に示すメイン側のアテンション・データ・レジスタからデータを受け取る．そののち，❼に示すアテンションアック・レジスタをアクセスすることで，サブシステムのメインアック・ビットがONに，またメイン・システムにかかっていた割り込みが解除され，もとの状態に戻る．

❹サブシステム・アテンションFIRQフラグ・レジスタ

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
サブシステム $AFE7 Access	アクセスするだけでメイン・システムへ割り込みをかける。							

❺メイン・システムFIRQフラグ・レジスタ

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
メイン・システム $FD04 Read						未使用		アテント 0:なし 1:あり

❻メイン・システム　アテンション・レジスタ

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
メインシステム $FD20 Read	アテンション・データ　＃0							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$FD21 Read	アテンション・データ　＃1							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
$FD22 Read	アテンション・データ　＃2							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

❼メイン・システム　アテンションアック・レジスタ

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
メインシステム $FD04 Write	ここにライトすることによって，サブのメインアックビットがONになり，メインにかかっていた割り込みが解除される。							

③ キャンセルIQR

サブCPUがメインのコマンドを実行中であっても，メインCPUは必要に応じてサブCPUにIRQ割り込みを発生させ，強制的にコマンドの実行を解除することができる．

❸に示すキャンセルIRQビットに1を書いて，そののち再び0に戻すとサブシステムにキャンセルIRQがかかる．

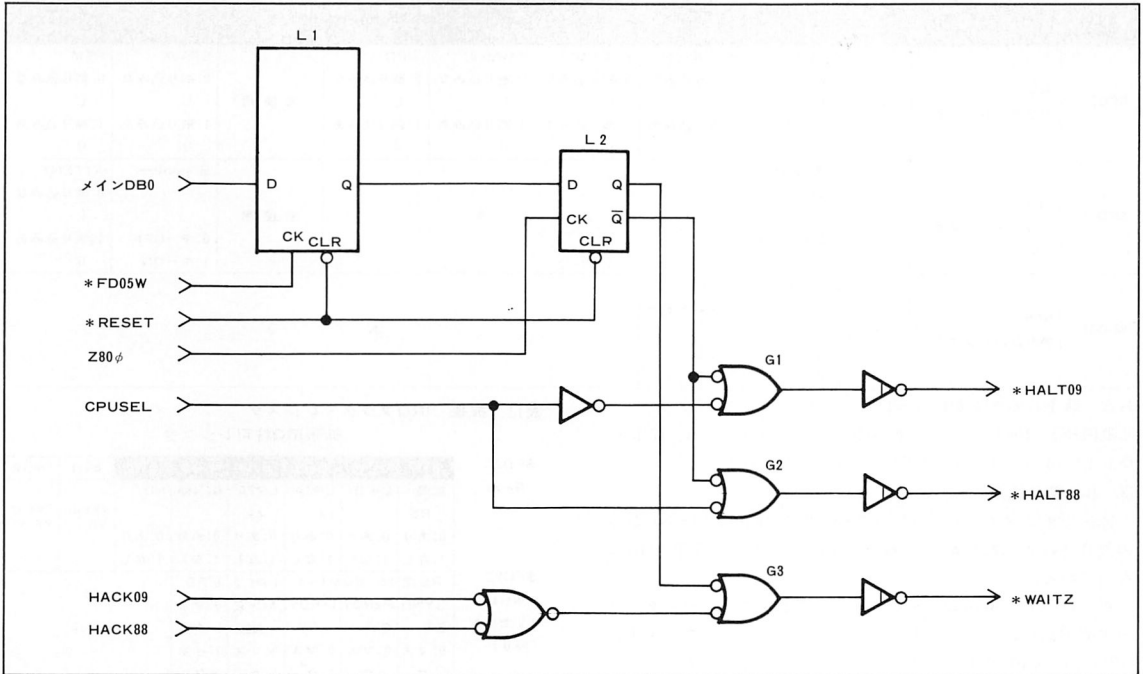
サブシステムのIRQ割り込みは，メインからのキャンセルの一系統のみなので，サブシステムはIRQ割り込みがかかるとすぐに，キャンセル処理を始めることができる．サブシステムにかかったIRQはサブアドレス＄AFE5，キャンセルIRQクリア・レジスタを読み出すだけで解除することができる．キャンセルIRQクリア・レジスタを❽に示す．

❽サブシステム・キャンセルIRQクリア・レジスタ

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
サブシステム $AFE5 Read	ReadすることでIRQを解除する							

メインCPU Z80CPUの切り換え

基本的にはメインCPUからZ80CPUへの切り換えは，＄FD05サブ，Z80インターフェイス・レジスタのビット0に1を書くことで，またZ80からメインCPUへの切り換えは0を書くことで

図14　CPU切り換えロジック

行なわれる．図14にCPU切り換えロジックを示す．

リセット直後はL2のQ出力は0であるので，*WAITZが有効であり，6809と8088の選別はCPUSELによって決まる．どちらにせよ，このままの状態ではZ80にwaitがかかり続け，Z80はアドレス0000Hで止まったままである．

メインCPUが$FD05のビット0に1を書くと，L1のQが1になり，これがZ80φでサンプルされてL2のQ出力となる．L2のQが1になると，メインCPUの*HALTがアクティブとなり，しばらくのちCPUが止まってHACK信号が1になりG4の出力が1になる．その結果，G3の出力が0となって*WAITZが解除となってZ80が動き出す．Z80からメインCPUに切り換えることもこれと同様にして行なわれる．

Z80CPUボードが実装されていないときは，Z80φが1になっているので，メインCPUが$FD05のビット0へ書き込んでもL2にその情報が伝わらず，メインCPUがそのために止まることはない．

Z80CPUボードは設計上，メインD-RAM，I/Oの$FD05，Z80CPUボード内のZ80用バンク・レジスタ，共有メモリ，i8251Aのリードおよびライトは保障しているが，それ以外のI/O，メモリはアクセスの保障をしていない．したがって，そのようなI/Oをアクセスするときは，いったん6809または8088CPUに制御を移し，I/Oをアクセスしてメモリにデータを展開してもらっておいてから，Z80に再び制御を戻し，このデータを使うという形式を取る．

また，CPUを切り換えたときに割り込みがはいると，CPUの切り換えタイミングによっては，双方のCPUが割り込み要求に応答しない（いわゆる〝抜け〟）か，あるいは双方共割り込み処理を行なってしまう（いわゆる〝重複〟）が発生する可能性がある．したがってCPUの切り換え時には，割り込みの発生を原則として禁止しなければならない．

Z80CPUはリセット後，0000Hからリスタートするようになっているため，メインCPUがこのアドレスにジャンプ命令を記込し，ユーザープログラムへ飛ばすようにしておく必要がある．

Z80用バンク・レジスタは自分自身はもとより，メインCPUからも書くことができる．このレジスタの内容を上位4ビットとしてZ80のアドレス$4000～$BFFFの領域をバンク・アドレスとする．表9にZ80用バンク・レジスタの構成を示す．

表9　Z80用バンク・レジスタ

		bit3	bit2	bit1	bit0
$FD2F Write (リセット時0)	あ　き	A_{19}	A_{18}	A_{17}	A_{16}

割り込み

6809CPUボード

6809CPUはNMI, IRQ, FIRQの3種の割り込みをサポートしていて，各割り込みは表10に示すアドレスからその割り込みもとがわかる．

① NMI割り込み

DMAタイム・オーバー：ホールト・バースト・モードのDMA転送時，I/OとDMAコントローラとの間でDMA転送のやりとりが正常に行なわれなかった場合，システムはデッドロックを起こしてしまうため，これを避けるために一定時間以上DMAが終了しない場合は，強制的にDMAコントローラにリセットをかけてDMAを終了させると共に，メインCPUにNMI割り込みを発生させてDMA転送が正常終了しなかったことを通知する．

拡張メモリ・パリティ割り込み：オプションの拡張メモリに付いている，パリティエラーで生じる割り込みである．割り込み許可レジスタのビット6を0にすることによって，このエラー検出機能を止めることができる．

② FIRQ割り込み

ATTENTION：サブCPUがメインCPUへ通知する割り込みで，インターバル・タイマ，クロック，プログラマブル・ファンクション・キーが要因である．

Breakキー：キーボードのBreakキーからの直接の割り込みで

表10　6809割り込みレジスタ

アドレス	内　容	R/W	ビット構成 bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bt 1	bit 0
$FD03	IRQ フラグ・レジスタ	R	RS-232C 0:割り込みなし 1:割り込みあり	キーボード 0:割り込みなし 1:割り込みあり	拡張IRQ 0:割り込みなし 1:割り込みあり	DMA(09) 0:割り込みなし 1:割り込みあり	5"FD 0:割り込みなし 1:割り込みあり	未使用	LPAcK 0:割り込みなし 1:割り込みあり	PTM 0:割り込みなし 1:割り込みあり
$FD04	FIRQ フラグ・レジスタ	R	外部FIRQ 0:割り込みなし 1:割り込みあり	あき				未使用	Breakキー 0:キーOFF 1:キーON	ATTENT 0:割り込みなし 1:割り込みあり
$FD57	NMI 割り込みレジスタ	R	拡張メモリパリティ 1:なし 0:あり	DMA タイム・オーバー 0:なし 1:あり	あき					

ある．離されるまでかかりっぱなしになっている．

拡張FIRQ：100ピンバスの＊FIRQ2がつながっている．拡張スロットにはいるモジュールからの割り込みである．

③　**IRQ割り込み**

RS-232C：メイン・ボード上のRS-232C（#0のポート）からの割り込みで，i8251AのTxRDY，RxRDY，SYNDETのORになっている．

キーボード：キーボード・エンコーダからのキーデータ・リード要求割り込み．キーボード・エンコード・データはメインCPU，サブCPUのどちらでも読み取りが可能であるが，以下の条件がある．

1)　キーボード割り込みビットが＄FD02のビット７にあり，メインCPUはこのビットに１を立てることにより，キーボードからの割り込みがメインCPUのIRQにはいるようになっている．電源リスタート直後はこのビットは０になっており，サブCPUがキーボードからの割り込みをFIRQで受け取るようになっている

2)　キーデータを読み取ることにより，レジスタ類は自動的にクリアされる．すなわち，データを２度読むことはできない．

拡張IRQ：100ピンバスの＊IRQ2がつながっている．拡張スロットモジュールからの割り込みである．

DMA：6809CPUボード内のDMACHD46504からの割り込みで，DMA転送終了のときにメインCPUにIRQをかける．

5″フロッピィっディスク：本体内のFDCがCPUに与えられたコマンドを終了したときに，割り込みを発生する．FDC内のステータス・レジスタのリードか次のコマンドのライトで復帰する．

LP Ack：ラインプリンタからのAck信号により割り込みがかかる．

ビット７以外の全ビットで，０のときは割り込みを禁止し，１のときに割り込みを許可する．

オプション・カード用のIRQフラグ，マスク・レジスタは，別にアドレス＄FD2Cに設けてある．**表11**に示す各ビットの内容については，各オプション・カードの仕様書を参考のこと．また，このレジスタと＄FD03の拡張IRQビットは無関係である．

8088CPUボード

8088CPUの割り込み入力はINTRとNMIの２本である．NMIは6809の場合と同様に，DMAのタイム・オーバーを見るのに使っているので，IRQとFIRQはすべてINTRで処理を行なうことになる．割り込みもとを知るためのレジスタは，**表10**よりDMAタイム・オーバーとDMA（09）のビットがなくなっただけで，残りの割り込みは6809CPUの場合とまったく同じである．また，割り込み許可レジスタも6809の場合とまったく同様に使える．

表11　拡張　IRQフラグ・レジスタ
拡張IRQ許可レジスタ

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$FD2C Read	拡張RS 0:あり 1:なし	8"FD 0:あり 1:なし	LP1Ack 0:あり 1:なし	LP2Ack 0:あり 1:なし	IE^{3}488 0:あり 1:なし	HD 0:あり 1:なし	reserve	reserve
$FD2C Write（リセット時０）	RS/232C COM1〜4 SYNDET 0:マスク 1:イネーブル	RxRDY 0:マスク 1:イネーブル	TxRDY 0:マスク 1:イネーブル	LP1,2 ACKENB 0:マスク 1:イネーブル	8"FD 0:マスク 1:イネーブル	reserve		

表12　8259AのIR入力と割り込み要因

割り込み入力	要　因	割り込み入力	要　因
IR0	タイマ	IR5	8087，拡張IRQ，ユーザIRQ
IR1	キーボード，Break	IR6	5"FD
IR2	(8"FD, HARD DISK)	IR7	LP Ack
IR3	ATENTION，拡張FIRQ ユーザFIRQ		
IR4	RS-232C		

8088CPUボードではIRQ，FIRQ割り込みのようなレベル割り込みのコントロールにMBL8259A-2を使っている．8259Aの割り込み入力と各要因との関係を**表12**に示す．

MBL8259AはMBL8086，8088周辺の割り込みコントローラで以下のような特徴を持つ．

●８レベルの割り込みをサポート．

●割り込みモードをプログラマブルに設定することができる．

●各割り込みごとにマスクができる．

8259Aの内部には，CPUがリードあるいはライトできるレジスタは９個あるが，8251Aと同様に書き込む順に従って各レジスタを選択しているため，CPUのI/Oアドレス上では２バイトのみ占める．

詳しくは8259Aのデータ・シート参照のこと．

サブシステム

サブシステムに使用しているMBL6809Eには３種の割り込み，NMI，IRQ，FIRQがあり，**表13**に示す目的に用いている．その他にRESET入力があるが，これはシステムのリスタートの際に用いる．

表13　サブシステム割り込み用途

割込名	用　途	そ の 他
NMI	サブシステム・インターバル・タイマ	20mSに1回かかる.
IRQ	キャンセル IRQ	メインからのコマンド・キャンセル要求. 要求のクリアはキャンセルIRQクリア・レジスタをリードする.
FIRQ	キーボード・ライトペン割り込み	キーボードあるいはライトペンからのデータ readyによる割り込み. 要求のクリアはFIRQクリア・レジスタをリード.

FIRQクリア・レジスタはライトペン割り込みかキーボード割り込みかを識別するビットがついていて，これによりどちらの割り込みがかかったかを区別する．どちらのビットも割り込みがあると〝1〞になる．また，このレジスタは読み出されるとその内容をクリアした上に，かかっていたFIRQが解除されるようになっている．

DMA

DMAコントローラ（以下DMACと略す）は各CPUボードに1つずつ載っており，6809CPUボードにはHD46504-2(MC68B44相当）が，8088CPUボードにはi8237A-5が使われている．したがって，DMAコントローラおよびバンク・レジスタは各CPUで独立のアドレスを持っている．

6809CPUボード使用時

機能

メイン・アドレス＄FD40～＄FD56にDMAC内のレジスタが，＄FD94～＄FD97にDMAバンク・レジスタがそれぞれ割り当てられている．さらに＄FD57にNMI割り込みレジスタ/DMA要求コントローラが割り当てられている．

DMA空間はCPUのウィンドウおよびMMRに依存しない絶対アドレス空間として扱われ，アドレスの下位16ビットをDMACが，上位4ビットをDMAバンク・レジスタが発生する．したがって，1回のDMA転送で動くデータの量は最大64Kバイトであり，かつ下位16ビットからの桁あふれを生じないようにDMACの設定，バンク・レジスタの設定を行なう必要がある．

割り込み

6809CPUボード使用時のみ，DMAC自身からIRQを発生できる．DMACからのIRQの発生は$FD03のIRQフラグ・レジスタを見ることで知ることができる．表14にIRQフラグレジスタを示す．

表14　IRQフラグ・レジスタ

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$FD03 Ready				DMA 0:割り込みなし 1:あり		未使用		

8088CPUボード使用時

機能

I/Oアドレス0010H～001FHにDMAC内レジスタが，0020H～0023HにDMAバンク・レジスタがそれぞれ割り当てられている．

DMA空間はアドレスの下位16ビットをDMACが，上位4ビットをDMAバンク・レジスタが発生する．したがって1回のDMA転送で動くデータの量は最大64Kバイトであり，かつ下位16ビットからの桁あふれを生じないようにDMACの設定，バンク・レジスタの設定を行なう必要がある．

8088CPUを使ったDMAではDMA中でもCPUが動くので，ソフトウェアで監視することでDMAによるデッドロックを防ぐことができる．ゆえに，6809の場合と違って，タイム・オーバー用のハードロジックを持っていない．

BASIC ROMカード(STのみ)

BASIC ROMカードはFM-11STにのみ標準実装され，EX, ADでの5″FDCカードが実装されるコネクタに実装して用いる．したがって，メイン・ブロックのアドレス・マップ上，5″FDCと同じアドレスに配置される．

BASIC ROMカードの制御は，BASIC ROMセレクト・レジスタとBASIC ROMデータ・レジスタの2つで行なわれ，セレクト・レジスタでカード内のROMのセレクトと内部カウンタのリセットを，データ・レジスタでROM内のデータの受け取りと内部カウンタのカウント・アップを行なう．表15，16に構成を示す．

表15　BASIC ROMデータ・レジスタ

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$FD1F Read	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

表16　BASIC ROMセレクト・レジスタ

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$FD1F Write	あ　き						ROM2 1：セレクト 0：非セレクト	ROM1 1：セレクト 0：非セレクト

BASIC ROMカードは，CPUがレジスタを介してROMのデータを受け取り，これをRAM上に展開して用いるというように使用する．したがってBASIC ROMカード内のROMにCPUの制御を渡すことはできない．

BASIC ROMカードの使用方法は次のようにして行なう．

①BASIC ROMセレクト・レジスタで必要なデータのはいっているROMを選択する．この際，内部のROMアドレス・カウンタがリセットされる．

②BASIC ROMデータ・レジスタより必要数だけデータを読み出す．データ・レジスタを1回アクセスするごとに，内部のROMアドレス・カウンタがインクリメントされる．したがって，CPUはデータ・レジスタを単に連続アクセスするだけで良い．

③BASIC ROMセレクト・レジスタで両方のROMを非セレクトにする．

アクセス・アドレスとディスプレイ・アドレス

FM-11のグラフィックVRAMは全部で192Kバイトあり，3バンクの複数ページという形でサブCPUは管理している．バンクおよびページに関しては表示系とアクセス系が分離しており，表示画面に使われているエリア以外のエリアをCPUが書き換えることが可能となっている．このため，画面の大幅な書き換えを，あたかも瞬時に行なったように見せることもできる．

表示系のバンク・ページのコントロール・ビットはサブアドレスの＄AFE2にあり，グラフィックVRAMディスプレイ・モード・レジスタと呼ぶ．構成を表17に示す．

表17　グラフィックVRAM

ディスプレイ・モード・レジスタの構成

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$AFE2 (Writeonly リセット時clear)	あき	モード設定ビット M1	あき	ページ・セレクト・ビット PS2	PS1	バンク・セレクト・ビット 0：表示しない 1：表示する RAM3	RAM2	RAM1

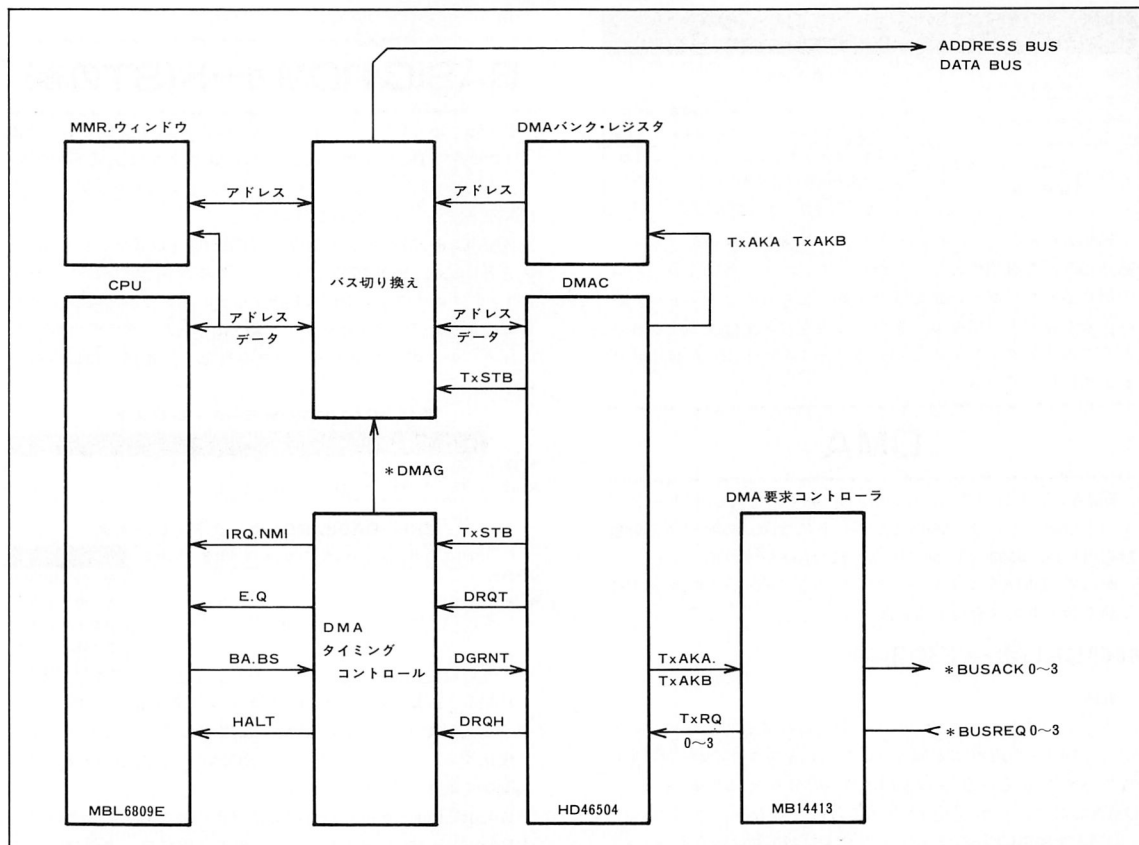



図15 DMA周辺ブロック図

図16 DMA周辺ブロック図

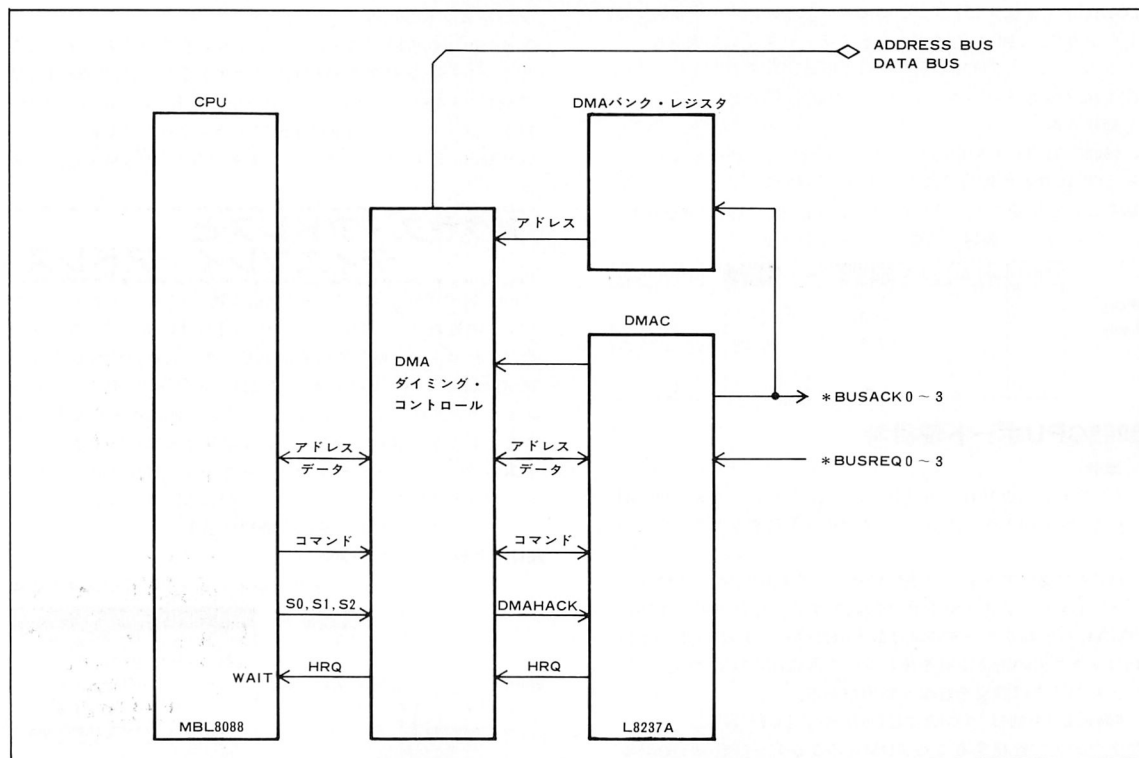
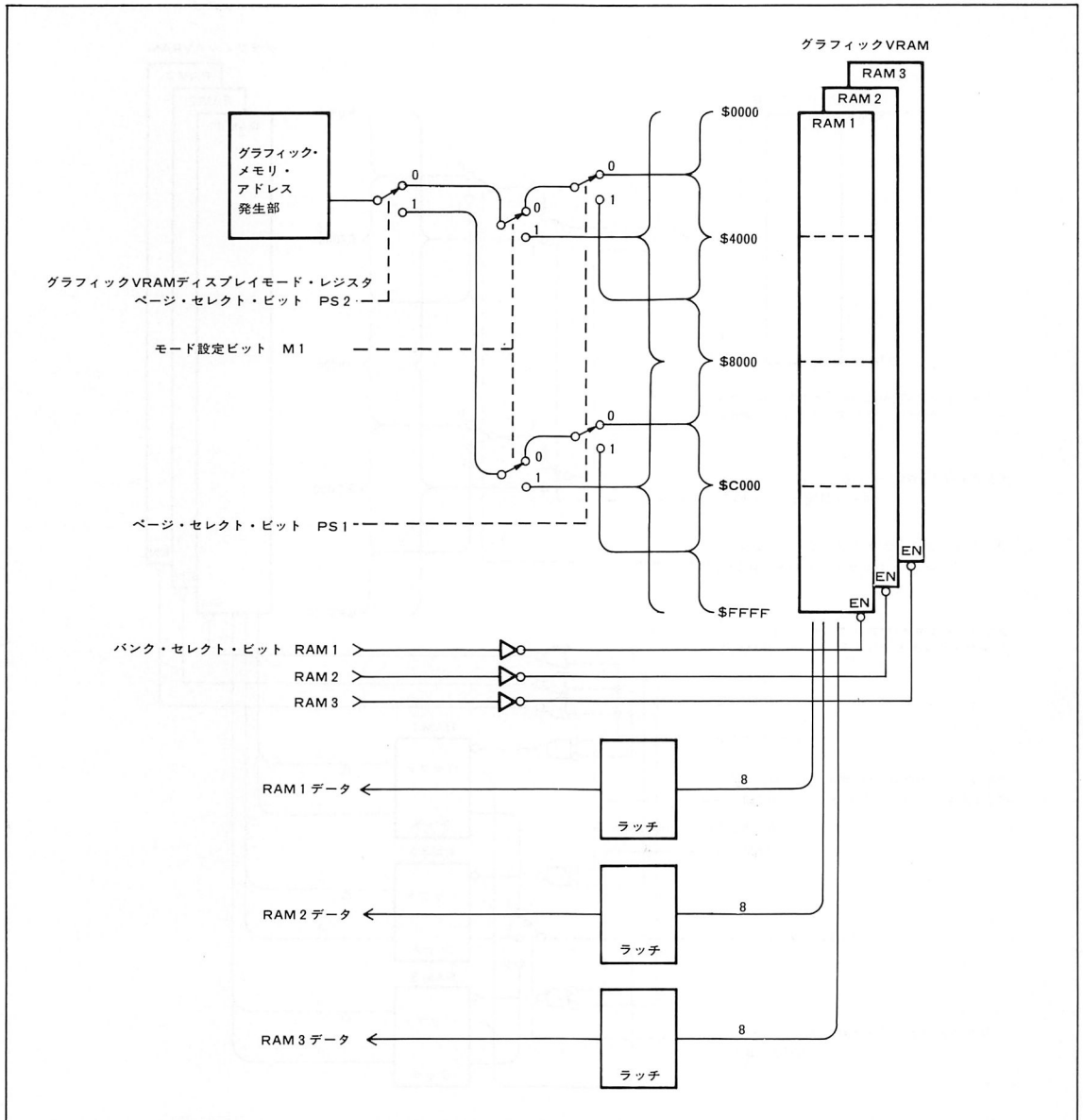



図17　表示系バンク・ページ　コントロール・ロジックとグラフィックVRAMアドレス

パレット・レジスタが初期状態にあるとき，RAM3～RAM1とグラフィックの色とは次の関係にある．

- Blueのドット　←　RAM1内のデータ
- Redのドット　←　RAM2内のデータ
- Greenのドット　←　RAM3内のデータ

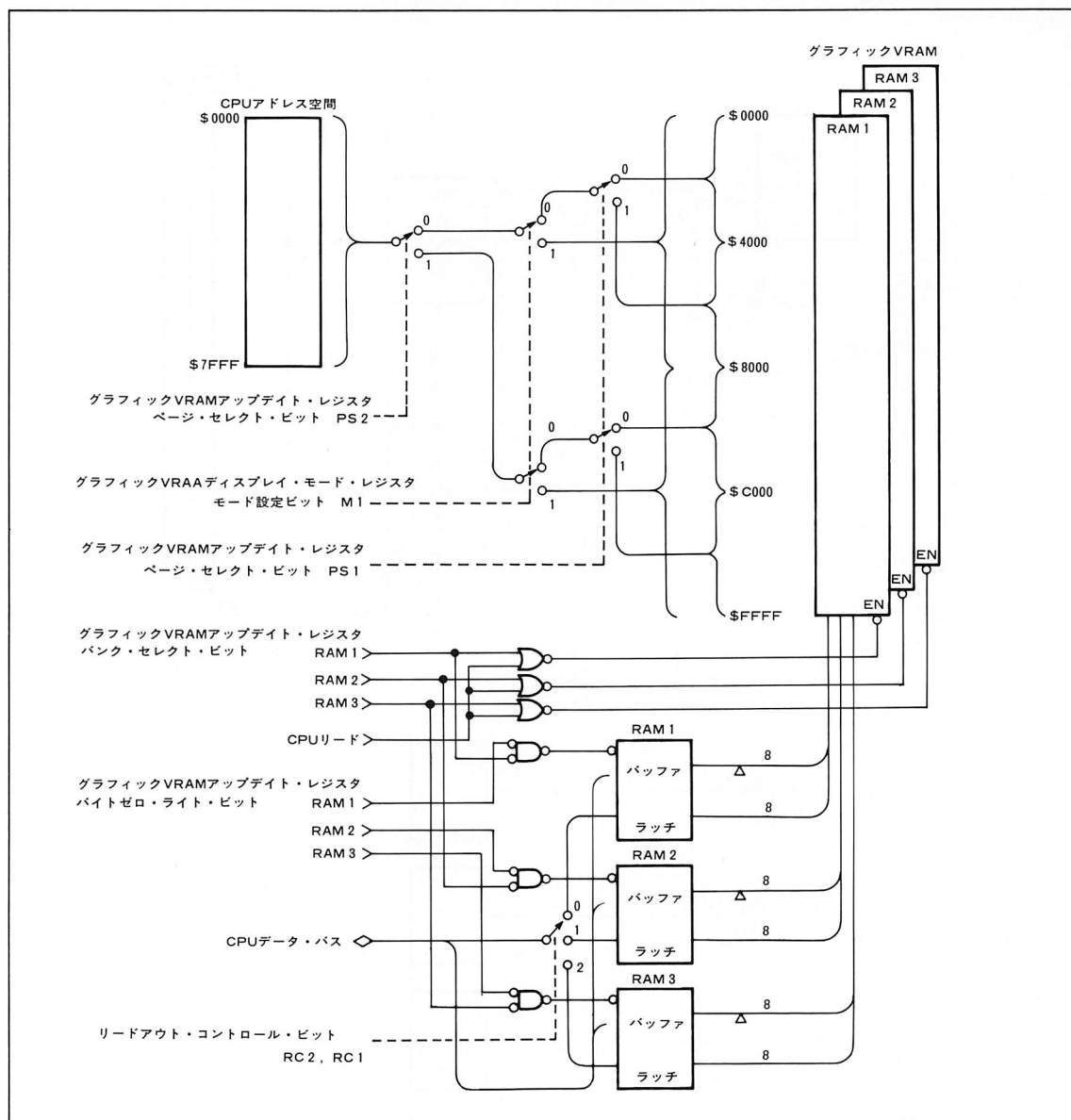
表示系のバンク・セレクト・ビットは，そのビットに該当するRAM内のデータを画面に表示するかしないかを決める（アクティブHigh）．例をあげると，RAM3のみ0でRAM2，1が1のときは，グラフィック画面はパレット・レジスタが初期状態にあるときは黒，青，赤，マゼンダの4色の表示になる．具体的な使い方としてはOHPのように表示図形の重ね合わせが瞬時に行なう場合などである．

ページ・セレクト・ビットは2ビットで1組の構成になっており，このビットの内容がグラフィックVRAMのアドレスの上位2ビットになる．ただし，モード設定ビットM1の値によっては，PS1の内容が無効になり，ページ選択はPS2のみで行なうようになる．ページ・セレクト・ビットとグラフィックVRAMに与えられるアドレスとの関係を**表18**に示す．

表18　ページ・セレクト・ビットとグラフィックVRAMのアドレスの関係

PS2	PS1 ＼ M1	0	1
0	0	$0000～$3FFF	$0000～$7FFF
0	1	$4000～$7FFF	$0000～$7FFF
1	0	$8000～$BFFF	$8000～$FFFF
1	1	$C000～$FFFF	$8000～$FFFF

モード設定ビットM1は400ドットのタイミング時に400ドットの表示にするか，擬似的に200ドットの表示にするかを決める．1にすると400ドットに，0にすると200ドットの表示になる（ただし，RAMの読み出しタイミングはどちらでも同じである）．このビットのみ，表示系とアクセス系が共通である．また，200ドットのタイミングでこのビットに1を書いたときの動作は保障しない．

図18　アクセス系バンク・ページ　コントロール・ロジックとCPUアドレス・グラフィックVRAMアドレス

アクセス系のバンク・ページ コントロール・ビットはサブアドレスの$AFE1と$AFE3にあり，それぞれグラフィックVRAMアップデイト・レジスタ，グラフィックVRAMページ・セレクト・レジスタと呼ぶ．

アクセス系のバンク・セレクト・ビットは，そのビットに該当するRAMへデータを書くかどうかを決める（アクティブHigh）．例をあげると，RAM3のみ0でRAM2,1が1のときは，CPUのライト動作でRAM2，RAM1に対応するRAMにデータが書き込まれ，RAM3は影響を受けない．このように1回のライト動作で最大3つのバンク内のデータを書き換えることができる．このビットはライト動作のみ関与し，リード時はこのビットが0のRAMの内容でも読み出すことは可能である．

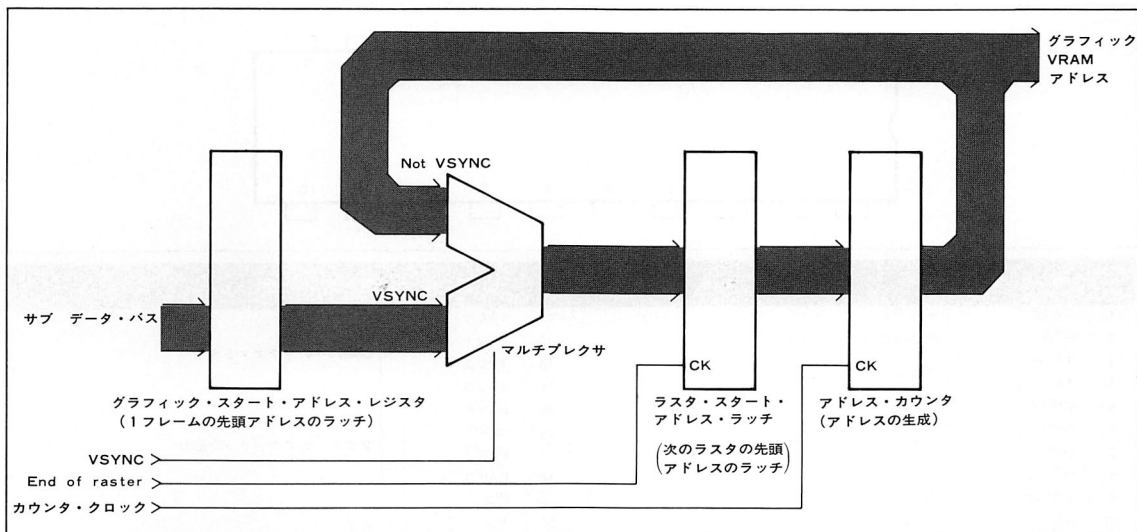
バイトゼロ・ライト・ビットはバンク・セレクト・ビットが0，つまりライト・セレクトされていないバンクに対して，他のバンクのライト動作と同時に0を書くように指定するビットである（アクティブLow）．用途としては，バイト・オリエンテッドに区切られた図形（漢字etc.）の書き換えなどがあげられる．

リードアウト・コントロール・ビットはグラフィックVRAMの，どのバンクを読み出すかを指定するビットで，2ビット1組になっている．リードアウト・コントロール・ビットとバンクの関係を**表19**に示す．

表19　リードアウト・コントロール・ビットとバンクの関係

リードアウト、コントロール		アクセスされるRAM
RC 2	RC 1	
0	0	RAM 1
0	1	RAM 2
1	0	RAM 3
1	1	データ・バス上のデータは不定

ページ・セレクト・ビットは2ビットで1組の構成となっており，このビットの内容がグラフィックVRAMのアドレスの上位2ビットになる．ただし，グラフィックVRAMディスプレイモード・レジスタのモード設定ビットM1の値によっては，PS1

図19　グラフィックVRAMアドレス生成部ブロック図

の値が無効になり，ページ選択はPS2のみで行なうようになる．グラフィックVRAMディスプレイモード・レジスタ中のページ・セレクト・ビットと異なる値にすることにより，画面表示用のエリアでないエリアにCPUが書くことができる．ページ・セレクト・ビットとグラフィックVRAMに与えられるアドレスとの関係は表示系と同じであり，**表18**がそのまま使える．

表示系のバンク・ページ コントロール・ロジックとグラフィックVRAMとの関係を**図17**に，アクセス系のバンク・ページ コントロール・ロジックとCPUアドレス・グラフィックVRAMとの関係を**図18**に示す．

FM-11のグラフィック画面はスクロールが可能になっているが，これは画面の左上の点に対応するメモリのアドレスをCPUが任意に設定できることによる．ただし，グラフィックVRAMディスプレイモード・レジスタとグラフィックVRAMアップデイト・レジスタで決められたエリアからはみ出すことはできない．これも表示系とアクセス系が独立していて，前者がグラフィック・スタート・アドレス・レジスタ，後者がアクセス・スタート・アドレス・レジスタと呼ぶ．

グラフィック・スタート・アドレス・レジスタは表示画面の左上の点に対するグラフィックVRAMのアドレスを規定する．したがって，この内容を徐々に増やすとグラフィック画面は左の方へずれる．逆に減らすと右の方へずれる．また，画面の途中でグラフィックVRAMのアドレスに桁あふれが生じることがあっても，現在表示中のエリアの初めのアドレスに戻って表示を続ける．モード設定ビットM1が0のときにはA_{14}の指定は無効になる．

アクセス・スタート・アドレス・レジスタはサブアドレス＄0000に対するグラフィックVRAMのアドレスを規定する．具体的には次の式のようになる．

グラフィックVRAMアドレス＝CPUのアクセス・アドレス＋アクセス・スタート・アドレス・レジスタ

このレジスタにより，グラフィックVRAM画面をスクロールしても，常に画面の左上の点をCPUアドレスの＄0000に設定することができる．加算の結果アクセス・エリアに桁あふれが生じても，あふれた分だけオーバーラップして，必ずもとのエリアにアクセスするようになっている．したがってモード設定ビットM1が0のときはA_{14}の指定は無効になる．

FM-11のグラフィックVRAMはD-RAMであるので，表示以外の期間もアドレスのカウント・アップをして，リフレッシュの保障をしている．そのため，1ラスタごとにカウンタの再設定を行なう必要がある．**図19**中のラスタ・スタート・アドレス・レジスタはアドレス・カウンタが発生した次のラスタの先頭アドレスをラッチし，次のラスタの開始時にアドレス・カウンタにロードさせることにより，この役目を果たしている．また，VSYNC期間中にグラフィック・スタート・アドレス・レジスタよりデータをもらうことで，フレームの開始時のスタート・アドレスをアドレス・カウンタに設定することも行なっている．**図19**はグラフィックVRAMのアドレス生成部のブロック図である．

キャラクタVRAM

キャラクタVRAMのアクセス・タイミング

キャラクタVRAMはキャラクタ・コードを保持するキャラクタVRAMと，その色調およびCRTへの表示状態を指定するアトリビュートVRAMから成る．

グラフィックVRAMと同様で，200ドットのタイミングと400ドットのタイミングでアクセス方法が異なる．また，キャラクタVRAMのスキャン・アドレス発生にCRTコントローラLSIを用いているため，高速のスクロールが可能となっている．

キャラクタVRAMとアトリビュートVRAM

キャラクタVRAMはCPUのアドレス空間の＄8000～＄8FFFの4Kバイトにあり，偶数の2Kバイトがキャラクタ VRAM，奇数の2KバイトがアトリビュートVRAMに分けられている．

キャラクタVRAMはキャラクタ・データを保持するRAMで，CPUは表示したい文字のコードをここへ書くことでキャラクタの表示が行なえる．

アトリビュートVRAMは，対応するキャラクタVRAMが表示する文字の表示の属性を決めるもので，アドレス$2n$のキャラクタVRAMに対応するアトリビュートVRAMのアドレスは$2n+1$である．アトリビュートVRAMの構成を**表20**に示す．

① bit0～2,5 Blue, Red, Red, Green, Intensity

この4ビットの指定により，キャラクタ・パターンのドット・フォントの立った部分に，16通りの色がつく．色調は**表21**に示す．

ただし，Intensityビットが影響するのはカラーCRTのみで，

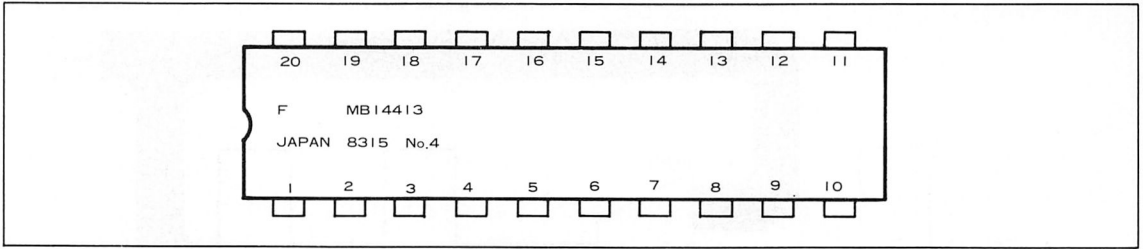
図20　MB144413のピンと機能

ピンNo.	信号名	用途	ピンNo.	信号名	用途
1	$\overline{\text{RESET}}$	リセット入力	11	TRQ 0	DMACへのリクエスト出力
2	$\overline{\text{STB}}$	DMA転送のストローブ入力	12	TRQ 1	
3	TAKA	DMACよりのアクノリッジ入力	13	TRQ 2	
4	TAKB		14	TRQ 3	
5	$\overline{\text{BRQ 0}}$	バス・リクエスト入力	15	$\overline{\text{BAK 0}}$	要求元へのアクノリッジ出力
6	$\overline{\text{BRQ 1}}$		16	$\overline{\text{BAK 1}}$	
7	$\overline{\text{BRQ 2}}$		17	$\overline{\text{BAK 2}}$	
8	$\overline{\text{BRQ 3}}$		18	$\overline{\text{BAK 3}}$	
9	ϕ_2DMA	DMACクロックの入力	19	NC	
10	GND	Ground	20	*Vcc*	＋5 V

表20　アトリビュートVRAMのビット構成

bit7	bit6	bit5	bit4	bit3	bit2	dit 1	bit0
		Intensity	Blink	Reverse	Green	Red	Blue
		1：ON 0：OFF	1：ON 0：OFF	1：ON 0：OFF	1：ON 0：OFF	1：ON 0：OFF	1：ON 0：OFF

表21　Intensity,Green,Red,Blueビットと表示される色の関係

bit5 Intensity	dit2 Green	bit I Red	bit 0 Blue	色
0	0	0	0	BLACK
0	0	0	I	BLUE
0	0	I	0	RED
0	0	I	I	MAGENTA
0	I	0	0	GREEN
0	I	0	I	CYAN
0	I	I	0	YELLOW
0	I	I	I	WHITE
I	0	0	0	DARK GREY
I	0	0	I	LIGHT BLUE
I	0	I	0	LIGHT RED
I	0	I	I	LIGHT MAGENTA
I	I	0	0	LIGHT GREEN
I	I	0	I	LIGHT CYAN
I	I	I	0	LIGHT YELLOW
I	I	I	I	LIGHT WHITE

グリーンCRTのコンポジット信号についてはこの影響を受けない．また，カラーCRTでもIntensity入力がないものも，やはりこの影響を受けない．したがってDARK GREYで文字を書いても，グリーンCRTには表示されないし，またIntensity入力のないカラーCRTでも表示されない．

② bit 3 Reverse

このビットに1を書くことによって，ドット・イメージの反転出力が得られる．その色調はbit5，2～0で指定された通りになる．たとえば，あるアトリビュートRAMに#＄0Fを書き込んだとき，そのアトリビュートが付髄しているキャラクタの表示はドット・フォントのある部分はバックカラーで，ない部分が白で表示される．

③ bit4 Blink　このビットに1を書くことで，対応するキャラクタ表示が0.64sec周期で点滅をする．

パレット・レジスタの機能は，CPUによって設定された8バイトのデータ・テーブルにより，グラフィックVRAMからの3ビットのデータをテーブル・ポインタとして用いることで，色

表22　ビデオ出力コントロール・レジスタの構成

	bit7	bit6	bit5	bit4	bit3	bit2	bit 1	bit 0
メイン・アドレス $FDA0 (writeonly リセット時0)	あき				グラフィック		キャラクタ	
					グリーン	カラー	グリーン	カラー
					0：出力しない 1：出力する	0：出力しない 1：出力する	0：出力しない 1：出力する	0：出力しない 1：出力する

表23　パレット・レジスタのライト・データと色の関係

色	ライトデータ	色	ライトデータ
BLACK	$×0	DARK GREY	$×8
BLUE	$×1	LIGHT BLUE	$×9
RED	$×2	LIGHT RED	$×A
MAGENTA	$×3	LIGHT MAGENTA	$×B
GREEN	$×4	LIGHT GREEN	$×C
CYAN	$×5	LIGHT CYAN	$×D
YELLOW	$×6	LIGHT YELLOW	$×E
WHITE	$×7	LIGHT WHITE	$×F

注)：×は0～Fのいずれでも良い。

調信号4ビットに変換するものである．したがって，16色の色を発生できるが，同時には8色までの表示となる．

〈パレットレジスタの例〉

パレット・レジスタがデフォルト状態であったとき，CPUが＄FD9Bに＄F6を書いたときLIGHT MAGENTAのグラフィック図形の色がYELLOWに変わる．これは初めRAM3＝0，RAM2＝1，RAM1＝1→＄FD9Bの内容の読み出し→I＝1，G＝0，R＝1，B＝1でLIGHT MAGENTAを出力していたものがI＝0，G＝1，R＝1，B＝0という読み出し値に変わり，YELLOWを出力するようになったためである．

パレット・レジスタに書き込む値と色の関係は**表23**に示す通りである．

ビデオ出力コントロールレジスタはキャラクタ，あるいはグラフィック・データをCRTに表示するかしないかを決定するもので，カラー，グリーンが独立に設定できる．したがって，カラーCRT画面にグラフィック図形とキャラクタを，グリーンCRTにキャラクタのみを表示させるということが，ソフトウェアで設定できる．また，キャラクタを消した方の画面ではキャラクタフォントの下に隠れていたグラフィック・フォントをも表示する．

図21　MB14413　内部等価回路

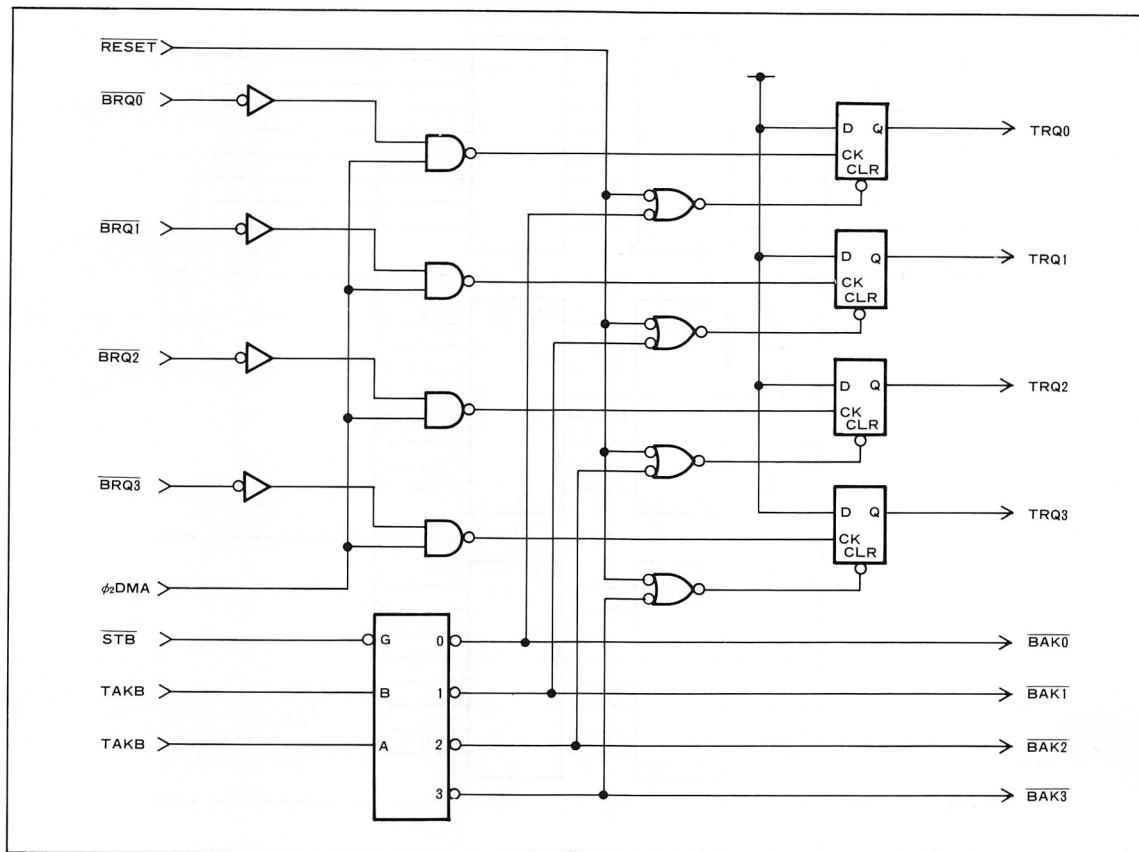
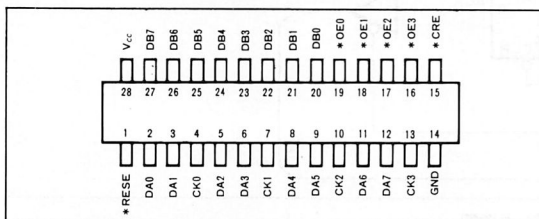



図22　MB14414　ピンと機能

カスタムLSI

FM-11にはメイン・ブロック，サブブロック合わせて5個のカスタムLSIが使われている．以下はその一覧である．

- **MB14413**（DMA要求コントローラ）
- **MB14414**（アテンション・レジスタ）
- **MB14416**（メイン・システム タイミング・ジェネレータ）
- **MB15021**（パレット・レジスタ）
- **MB15022**（サブシステム タイミング・ジェネレータ）

DMA要求コントローラMB14413

このLSIはDMA転送要求信号を，HD46504DMACの入出力規格に合わせるように，タイミングを取るためのものである．**図20**にピン配置図，**図21**に内部等価回路を示す．

アテンション・レジスタMB14414

このLSIはアテンション機能を果たすもので，サブ側から書き込み専用，メイン側から読み出し専用の3バイト・レジスタ，メイン側から書き込み専用，サブ側から読み出し専用の1ビットのFFがはいっている．**図22**にピン配置を，**図23**に内部等価回

図25　MB14414　ピンと機能

ピンNo.	信号名	用途
1	※RESET	リセット
2	DA0	サブ側データ・バス
3	DA1	
4	CK0	サブ側アテンション・レジスタ＃0ライト・パルス入力
5	DA2	サブ側データ・バス
6	DA3	
7	CK1	サブ側アテンション・レジスタ＃1ライト・パルス入力
8	DA4	サブ側データ・バス
9	DA5	
10	CK2	サブ側アテンション・レジスタ＃2ライト・パルス入力
11	DA6	サブ側データ・バス
12	DA7	
13	CK3	メイン側アック・トリガ入力
14	GND	Ground
15	※CLR	サブ側アテンションFIRQ要求入力
16	※OE3	サブ側アックビット・リード・パルス入力
17	※OE2	メイン側アテンション・レジスタ＃2リード・パルス入力
18	※OE1	メイン側アテンション・レジスタ＃1リード・パルス入力
19	※OE0	メイン側アテンション・レジスタ＃0リード・パルス入力
20	DB0	メイン側データ・バス
21	DB1	
22	DB2	
23	DB3	
24	DB4	
25	DB5	
26	DB6	
27	DB7	
28	Vcc	＋5V

図23　MB14414　内部等価回路

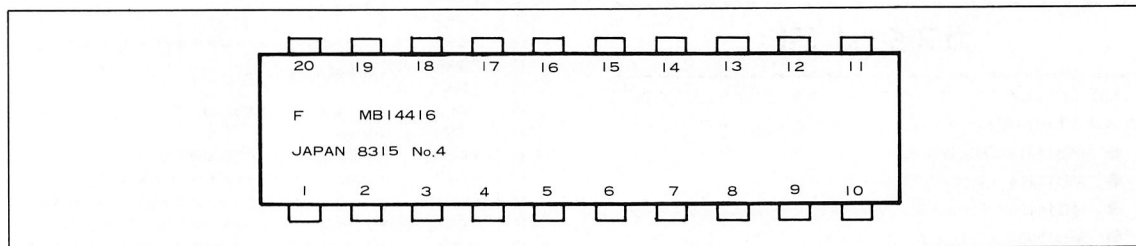
図24　MB14416ピンと機能

ピン No.	信号名	用途	ピン No.	信号名	用途
1	16MHz	16Mzクロック入力	11	$\overline{\text{REFRESH}}$	リフレッシュ出力
2	$\overline{\text{PONRST}}$	パワーオン・リセット入力	12	$\overline{\text{TSCGRNT}}$	TSC DMAの許可出力
3	HACK09	HALTDMAの許可入力	13	$\overline{\text{Q}}$	システム＊QB用出力
4	$\overline{\text{RESET}}$	リセット入力	14	CUUQ	CPU用Q出力
5	$\overline{\text{DRQT}}$	TSCDMA要求入力	15	CPUE	CPU用E出力
6	$\overline{\text{FCFD}}$	1μsのwait要求入力	16	$\overline{\text{CPUE}}$	システム＊EB用出力
7	$\overline{\text{TxSTB}}$	DMACのTxSTB信号入力	17	ϕ_2DMA	DMACのクロック出力
8	ϕcMRDY	メモリ・レディ入力	18	$\overline{\phi_2\text{DMA}}$	DMA時のシステム＊EB用出力
9	ϕc	バス上のCPUクロック用出力	19	$\overline{\text{DMAQ}}$	DMA時のシステム＊QB用出力
10	GND	Ground	20	V_{cc}	＋5Ｖ

図25　MB14416　内部等価回路

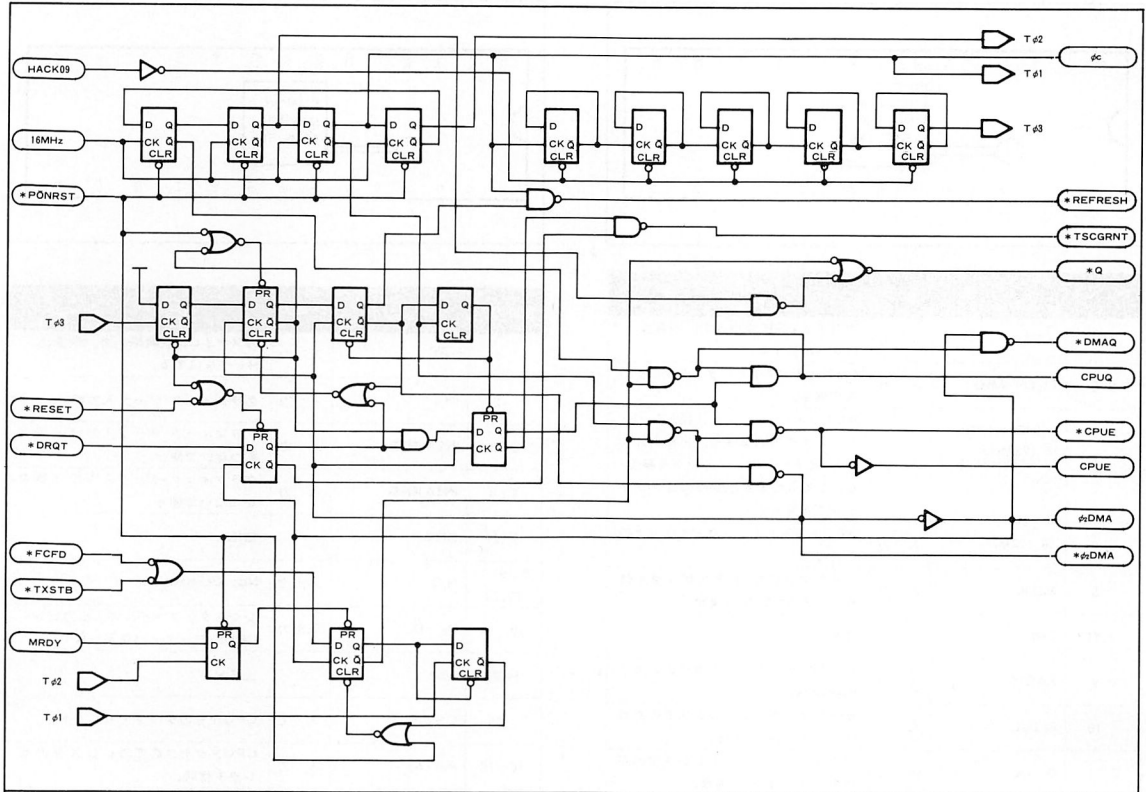



図26　MB15022の内部ブロック図

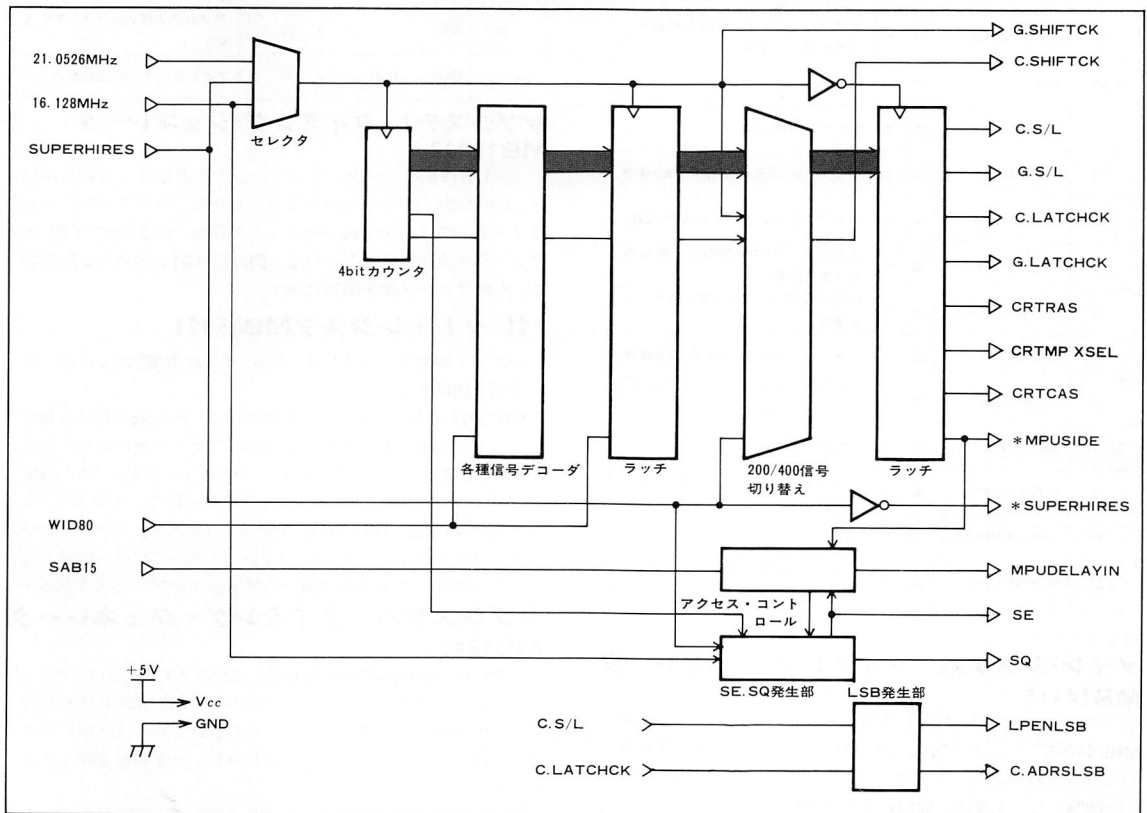
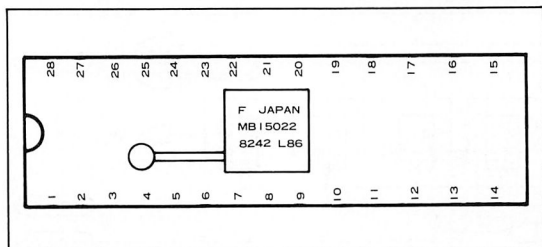
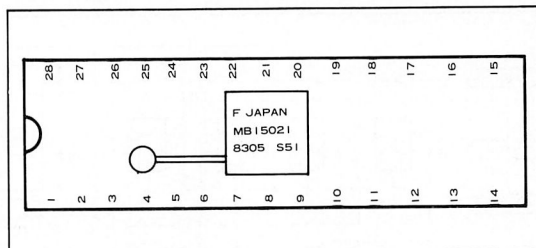



図27　サブシステム・タイミング・ジェネレータ

Pin No.	名　　称	入出力	用　　　途
1	*TEST	入　力	カスタムLSI試験用入力，通常はHレベルにする．
2,6	21,0526MHz	入　力	400ドット・タイミング・クロックの入力端子．
3	SUPERHIRES 16,128MHz SUPERHIRES	入　力	200ドット・タイミングと400ドット・タイミングの切替信号．Hを入力すると400ドットのタイミングを発生し，Lを入力すると200ドットのタイミングを発生する．
6,8	16,128MHz	入　力	200ドット・タイミング・クロックの入力端子．
5	WID80	入　力	キャラクタ40字/80字の切り替え信号，H入力で80字を指定．
7,21	GND		接地
9	SAB15	入　力	VRAM領域のデコード信号の入力，Highactve．
10	C. S/L	出　力	キャラクタ用シフトレジスタのためのシフト/ロード信号．
11	G. S/L	出　力	グラフィック用シフトレジスタのためのシフト/ロード信号．
12	C.LATCHCK	出　力	キャラクタ・データのためのラッチのクロック信号．
13	G. LATCHCK	出　力	グラフィック・データのためのラッチのクロック信号．
14,28	Vcc		＋5 V．
15	LPENLSB	出　力	ライトペン用LSB.
16	CRTRAS	出　力	グラフィックVRAM表示用RAS信号．
17	C. ADRSLSB	出　力	キャラクタ用表示アドレスのLSB．
18	CRTMPXSEL	出　力	グラフィックRAMのMPXの表示用切り替え信号．
19	MPUDELAYIN	出　力	CPUのVRAMアクセス時のタイミング信号．
20	CRTCAS	出　力	グラフィックVRAM表示用CAS信号．
22	*MPUSIDE	出　力	VRAMアクセスのタイミング・クロック，LでCPUを示す．
23,24	SE, SQ	出　力	CPUのE,Q信号．
25	C. SHIFTCK	出　力	キャラクタ用シフトレジスタのシフトクロック．
26	*SUPERHIRES	出　力	SUPERHIRESの反転信号．
27	G. SHIFTCK	出　力	グラフィック用シフトレジスタのシフトクロック．

路をそれぞれ示す．

メイン・システム　タイミング・ジェネレータ MB14416

このLSIは6809CPUボードの中にあって，メイン・ブロックの**MBL6809E**クロックの発生，MRDYのコントロール，HALT，TSC DMA時のクロックのサポート，リフレッシュ制御を行なう．**図24**にピン配置図，**図25**に内部等価回路を示す．

図28　パレット・レジスタのピンと機能

Pin No.	名　　称	入出力	用　　　途
1	$\overline{\text{INIT}}$	入　力	カスタムLSI試験用入力，通常はHレベルにする．
2	IN 1	入　力	グラフィック・データ入力ピン．
3，6	IN2A,IN2B	入　力	グラフィック・データ入力ピン，両者をつないで使う．
4，5	IN3A,IN3B	入　力	グラフィック・データ入力ピン，両者をつないで使う。
7，21	GND		接地．
8，9，22,23	N,C.	入出力	Non Connection.
10～13	D_0～D_3	入出力	レジスタ・ファイルの入出力ピン，CPUのデータ・バスとつなぐ．
14,28	Vcc		±5 V．
15	CS	入　力	CPUからのチップ・セレクト．
16～18	A_0～A_2	入　力	CPUアクセスの際のレジスタのセレクト信号．
19	$\overline{\text{RD}}$	入　力	CPUアクセスの際のリード・タイミング信号．
20	$\overline{\text{WR}}$	入　力	CPUアクセスの際のライト・タイミング信号．
24～27	OUT 1～OUT 4	出　力	グラフィック・データ出力ピン．

サブシステム・タイミング・ジェネレータ MB15022

このLSIはキャラクタ，グラフィックの表示コントロール信号，GVRAMのアクセスコントロール信号，ライトペン，キャラクタ表示アドレスの最下位ビットの作成，およびサブCPUのクロック生成を目的としている．**図27**に**MB15022**のピン配置図に，内部ブロック図を**図26**に示す．

パレット・レジスタMB15021

パレット機能を果たすLSIである．ピン配置**図28**に，内部ブロック図を**図29**する．

MB15021の入出力は２つの系統で，CPU系と表示系に分類することができる．表示系はレジスタ・ファイルを読み出すのみであり，IN1～3が入力で，OUT1～4が出力になる．一方，CPU系はレジスタ・ファイルにリード/ライト可能で，アクセス・コントロールにCS，WR，RD，A0～2入力があり，ファイルのデータはD0～3で読み書きする．IN1～3とA0～2が同一の値のとき，レジスタ・ファイルの同一のアドレスがアクセスされる．

サブシステム・タイミング・ジェネレータ MB15023

このLSIは**MB15022**の改良版として設計されたもので，200ドット・モードのときのRASのプリチャージ時間の不足を改善し，同相のクロックをシフトロードおよびクロックに用いることで，外付として必要であった*CR*のディレイ回路を除くことができる．

ピン配置と機能は，６，８ピンが入力からN．C．に変わった以外は**MB15022**と同じであり，これと差し換える形になる．

図29　パレット・レジスタ　MB15021の内部ブロック図

マザーバス(1)

ADDRESS BUS
DATA BUS
+5V
NC
*PONRST
*IOIHB
*QB
*REFCK
*RESET
*RAMIHB
*EB
RWB
AB1 AB3 AB5 AB7 AB9 AB11 AB13 AB15 AB17 AB19
DB1 DB3 DB5 DB7
AB0 AB2 AB4 AB6 AB8 AB10 AB12 AB14 AB16 AB18
DB0 DB2 DB4 DB6
CN3A
CN2
CN1
R47
RB4
RB5
RB6
RB7
RB8
RB9
R101 1k
6.8kΩ

マザーバス(2)

NC
*IRQ 1
*IRQ 3
*IRQ 7
*FIRQ 1
HACK 09
HACK 88
CPUSEL
Z 80φ
− 12 V
+ 12 V
CN3B
CN2
CN1
RB 9
RB 11
RB 13
+5V
6.8k
*BUSREQ 0
*BUSACK 0
*IRQ 0
*IRQ 2
*IRQ 4
*IRQ 8
*FIRQ 0
*FIRQ 2
*HALT 09
*HALT 88
*WAITZ
SYSCLKB
RB 8
RB 12
RB 14

メイン・アドレス・デコード

メインD-RAMバッファ

メインD-RAM

M141
MB8265-15
M192
M152
M201
M164
M204
M178
M216
+5V
C100
C105
CB101
C102
C106
C108
C107
0.1μ
RAS
CAS
WE
RFSH
DIN
DOUT
DI0
DO0
DI1
DO1
DI2
DO2
DI3
DO3
DI4
DO4
DI5
DO5
DI6
DO6
DI7
DO7
*RAS
*CAS0
*WE0
*RFSH
RAM ADDRESS
RAM OUT DATA
RAM IN DATA.
*CAS1
*WE1

M142
MB8265-15
M153
MB8265-15
M165
MB8265-15
M179
MB8265-15
M193
MB8265-15
M202
MB8265-15
M205
MB8265-15
M217
MB8265-15
+5V
RAS
CAS
WE
RFSH
DIN
DOUT
Vcc
Vss
C108
C109
C110
C111
C112
C113
C114
C115
0.1μ
RA 0
RA 1
RA 2
RA 3
RA 4
RA 5
RA 6
RA 7
DI0
DI1
DI2
DI3
DI4
DI5
DI6
DI7
DO0
DO1
DO2
DO3
DO4
DO5
DO6
DO7
*RAS
*CAS1
*WE1
*RFSH
RAM ADDRESS
RAM OUT DATA
RAM IN DATA

プリンタ/オーディオカセット・インターフェイス

RS-232C

プログラマブル・タイマ

スイッチ/ミニフロッピーインターフェイス

タイミング・ジェネレータ

S-RAM
S.DATABUS
DATABUS
M218
MB74LS245
M238
MB8128-15
M252
MB8128-15
MB74LS32
M229
MB74LS32
MB74LS244
M251
M250
MB74LS00
M156
MB74S04
MB74LS157
M237
MB74S51
M89
M88
MB74S08
M99
MB74S04
M112
MB74LS00
S.ADDRESSBUS
ADDRESSBUS
SUBG
*SUBSFL
*RWB
S.WRITE
*SRWEQB
*SYSRAM
EB
RWB
*SHALTAC
*SEORQB
SEB

キャラクタROM

S.ADDRESSBUS
S.DATABUS
MBM2732A-25
SAB11
SAB10
SAB9
SAB8
SAB7
SAB6
SAB5
SAB4
SAB3
SAB2
SAB1
SAB0
CE
OE/Vpp
*SC8
Vcc
GND
+5V
CB19
3.3μ
SDB7
SDB6
SDB5
SDB4
SDB3
SDB2
SDB1
SDB0
SN74LS273
M211
SN74LS273
M212
D
Q
CK
CLR
ATAD14
ATAD13
ATAD12
ATAD11
ATAD10
ATAD9
ATAD8
ATAD7
ATAD6
ATAD5
ATAD4
ATAD3
ATAD2
ATAD1
ATAD0
ASTAD
*SRWEQB
*CGSEL
*ASTADH
*ASTADL
*SRESETB

キーボード・インターフェイス

ライトペン・インターフェイス

サブI/Oデコーダ(1)

*MIXGVU
*GVDVOC
*CANINT
*ATENST
*ASTAD
*GSTAD
*KEYIN
*CRTCCS
*SQB
*KROMR
*ATTENTR
*ATTENTW
*KROMW
*BUZZOFF
*BUZZON
*ATENT#0
*ATENT#1
*ATENT#2
*ATENT#3
M255
MB74LS32
M254
MB74LS04
M257
SN74S138
M256
SN74S138
M248
MB74LS155
M258
SN74S30
R93
3.3kΩ
+5V
N.C
SAB11
SAB10
SAB9
SAB8
SAB7
SAB6
SAB5
SAB3
SAB2
SAB1
SAB4
SAB0
S.ADDRESS BUS
*SEORQB
*IOUSRAM
SQB
S.READ
*ATTENTW
*ATTENTR

サブI/Oデコーダ(2)

SQB
*GVDVOC
S.READ
SAB0
SEB
*CANINT
MB74LS04
M254
MB74LS155
M243
*GVDMSEL
*GVPAGE
*SFIRQCLR
*CANIRQ
*ASTAD
*SQB
*GSTAD
MB74LS155
M233
*ASTADH
*ASTADL
*GSTADH
*GSTADL
SEORQB
*ATENST
M246
MB74LS00
MB74LS139
M242
*STSEL
*ATTENT
*S.RESETB
*BUZZOFF
*BUZZON
M247
MB74LS74A
BUZON
KTYINH
KEYINL
MB74LS139
M242
*KEYIN

サブレジスタ(1)

S.WRITE
S.DATABUS
DATABUS
＊ATENT＃0
＊ATTENT0
＊ATENT＃1
＊ATTENT1
＊ATENT＃2
＊ATTENT2
＊ATNACK
MB74S04
S.READ
M156
MB14414
M206
SDB0 SDB1 SDB2 SDB3 SDB4 SDB5 SDB6 SDB7
DA0 DA1 DA2 DA3 DA4 DA5 DA6 DA7
DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7
＊ATTENT＃3
＊ATTENT
CK0 OE0 CK1 OE1 CK2 OE2 CK3 CLR OE3 RESET
Vcc
GND
+5V
CB11
0.1μ
MB74LS174
SDB6 SDB4 SDB3 SDB2 SDB1 SDB0
D
Q
CK
CLR
＊GVDMSEL
S.DATABUS
＊S.RESETB
MODEBIT1
PAGESEL2
PAGESEL1
RAM3EN
RAM2EN
RAM1EN
CBLINK
D2 IS468
+5V
R97
22k
MB84040B
RES Q9
CK Q10
M122
D3 IS468
MB74LS14
TIMCK
M144
R62
M144
MB74LS14
R60
M131
MB84040B
RES
CK Q5
MB74LS14
M144
MB74LS00
M166
BLINK
＊SNMI
TP24

サブレジスタ(2)

*SBUSY
CURDSLSB
*LDENABLE
WID80
*FLASH
INSLED
PS2
PS1
RDOUT2
RDOUT1
*ZEROWR3
*ZEROWR2
*ZEROWR1
RAM3
RAM2
RAM1
MB74S04
M32
MB74S04
M156
M123
MB74LS74A
+5V
MB74LS08
N.C.
MB74LS174
M196
MB74LS367A
M184
MB74LS244
M158
SN74LS273
M169
SDB0
M209
MB74LS74A
MB74LS155
SDB4
SDB3
*SHALTAC
S.DATABUS
*S.RESETB
*GVPAGE
S.READ
SQB
*MIXGVU
SAB 0
S.WRITE
*SEORQB

CRTコントローラ

MR.ADDRESSBUS
R.ADDRESSBUS
LDISPTMG
HSYNC
＊HSYNC
VSYNC
＊VSYNC
＊DISPTMG
DISPTMG
＊C.ADRSLSD
CUDISP
S163LOAD
＊G.DISPEN
RASTCLK
MB74LS175
M113
SN74LS273
M101
SN74LS273
M102
MB74LS175
M69
M112
MB74LS00
M57
M111
MB74LS04
M48
SN74S86
MB74LS51
M36
+5V
HD68B45S
M81
MB74S04
MB74S51
M89
M99
MB74S04
CB8
3.3μ
TP21
M80
MB74LS74A
M121
MB74LS02
M111
MB74LS04
S.DATABUS
SAB0
＊CRTCCS
S.READ
LPSTB
SEB
＊C.ADRSLSB
SUPERHIRES
＊S.RESETB
C.ADRSLSB
＊VSYNC
＊HSYNC
＊SUPERHIRES
MODEBIT1
LDISPTMG
DISPTMG
CUROSLSB
WID80
LPENLSB

スタートアドレス・レジスタ

D-RAMアドレス
GM14N
G.MEMAD
G.MPXAD
SN74S158
SN74S163
SN74S283
SN74S74N
MB74S08
MB74S04
MB74LS04
MB74S51
G.NEXTADRS
PS 2
MODEBIT1
PS1
S.ADDRESSBUS
A STAD
CRTRAS
S163LOAD
※MPUSIDE
G.LATCHCK

サブD-RAMタイミング

キャラクタRAM

キャラクタ・ジェネレータ

P-S変換
G.GREENDOT
G.RAM1DOUT
G.LATCHCK
G.REDDOT
RAM1EN
G.RAM2DOUT
G.BLUEDOT
RAM2EN
G.RAM3DOUT
WHITE
RAM3EN
G.SHIFT *LOAD
COLOR.CHENABLE
G.SHIFTCK
SW.CHENABLE
C.DOTDATA
AT2
AT1
AT0
DISPRDY
GR.BWEN
GR.COLEN
LBLINK
SN74LS273 M62
SN74LS273 M63
SN74LS273 M64
O17 O16 O15 O14 O13 O12 O11 O10
O27 O26 O25 O24 O23 O22 O21 O20
O37 O36 O35 O34 O33 O32 O31 O30
MB74LS166 M50
MB74LS166 M51
MB74LS166 M52
SN74S374 M26
GINT
CINT
M46 MB74LS32
M46 MB74LS32
M38 MB74S10
MB74S10 M38
M32 MB74S04
MB74S04
MB74LS51 M24
MB74LS51 M30
MB74LS51 M36
MB74LS51 M25
MB74LS51 M31
MB74LS51 M37
MB74LS51 M24
M66 SN74S32
M53 MB74S10
G.RAM1DOT
G.RAM2DOT
G.RAM3DOT
COLOR.GREEN
COLOR.RED
COLOR.BLUE
BW.GREEN
BW.RED
BW.BLUE
INTENSITY

CRTインターフェイス

漢字ROMコネクタ

ライトペン，キーボード・コネクタ

CN18
LIGHT PEN
G1
G2
TP13
※LPSW
TP12
※LPOUT
FB8
+5V
CN14
KEY BOARD
STOP
TP6
DATA
FB9
TP5
CLOCK
FB10
※KRESET
INSLEDON
G1
G2

マザーボード
RB10
RB6
RB7
RB8
RB9
RB1
RB2
RB3
RB4
RB5
CN8
CN7
CN6
CN5
CN4
CN3A
CN3B
6.8k
C4 0.1μ
C6 0.1μ
C1 100μ
C3 100μ
C5 0.1μ
C2 100μ
+5V
−12V
+12V
GND

CPU周辺

M49
MC68B09E
*RESETB
CPUE
CPUQ
*NMI
*FIRQ
*IRQ
*GOHALT
*DRQH
M61
MB74LS08
BA
BS
CPURW
R11
R1
1.5k
CB28
3.3
M50
MB485
CB29
RB6
M21
MB487
LOCAL ADDRESS BUS
M33
MB74LS245
RB2
RB5
M59
MB74LS244
CB30
0.1
DATA BUS
MAIN ADDRESS BUS
CPURW
*Q
CPUE
*CPUG
M26
MB74LS14
M60
MB74LS32
R2
560
C4
100p
3.3k

クロック/システム制御信号

MEMORY MAP RAM

DMAC周辺

M30
MC68B44
DGRNT
RES
φ2DMA
DRQH
DRQT
IRQ/DEND
Vcc
GND
TXRQ3
TXRQ2
TXRQ1
TXRQ0
TXAKA
CS/TXAKB
TXSTB
M19
MB74LS670
M9
MB74LS244
M47
MB74LS244
M46
MB74LS245
M20
MB74LS245
M56
MB74LS08
M23
SN7407
M25
MB74LS04
M32
MB74LS02
M53
MB74LS00
M43
TC4528BP
M54
MB74LS74A
M62
MB74LS125A
M26
MB74LS14
RB1
RB3
RB4
＊DRQH
＊DRQT
＊IRQ3
DATA BUS
＊DMACRST
＊NMI1
＊TXSTB
DGRNT
＊RESETB
＊DMACRST
φ2DMA
＊DMASEGST
TXAKA
TXAKB
MAIN ADDRESS BUS
＊DMAG
RWB
＊DMACS
＊NMISTRD
+5V
3.3kΩ

マザーバス・コネクタ(1/2)

MAIN ADDRESS BUS

CN1	Signal	CN1	Signal
7	*RESET	8	NC
9	AB0	10	AB1
11	AB2	12	AB3
13	AB4	14	AB5
15	AB6	16	AB7
17	AB8	18	AB9
19	AB10	20	AB11
21	AB12	22	AB13
23	AB14	24	AB15
25	AB16	26	AB17
27	AB18	28	AB19
29	TP5 *RAMIHB	30	TP2 *IOIHB
33		34	DB1
35	DB0	36	DB3
37	DB2	38	DB5
39	DB4	40	DB7
41	DB6 TP8 *EB	42	TP3 *QB
43	TP6 RWB	44	MRDY
45	NC	46	NC
47	NC	48	TP1 *REFCK

DATA BUS

マザーバス・コネクタ(2/2)

レディ回路

*DL 2 CK
*DL 3 CK
CPURDY
DL 8 CK
DMARDY
*DMADL 3 CK
ADDRESS BUS
M37 MB74S04
M21 MB74LS04
M57 MB74S10
M9 MB74LS10
M35 MB74LS00
M46 MB74LS08
M46 MB74LS08
M28 MB74LS00
M27 MB74LS164
M36 MB74LS164
M35 MB74LS00
M26 MB74LS20
M20 MB74LS27
M35 MB74LS00
M20 MB74LS27
M45 MD74LS245
C 9 0.1μF
+5V
AB19 AB18 AB17 AB16
DA 19 DA 18 DA 17 DA 16
TP4 TP2 TP3 TP5
*MEMR
*MEMW
*IOR
*IOW
CPUCK 88
ALE
HACK 88
DMAAEN
MRDY
*INTA
*DMAMEMR
*DMAMEMW
*DMAIOR
*DMAIOW
DMA HI-ADDRESS
DMACK 88
ADSTB
*DMAAEN

リフレッシュ回路

*REFCK
REFHRQ
CPUCK 88
DMACK 88
*DMACK 88
M48 MB487
M38 MB74LS02
M38 MB74S04
M30 MB74LS164
M39 MB74LS74A
M39 MB74LS74A
+5V
M21 MB74LS04
M59 MB487
M49 SN74LS393
M49 SN74LS393
TP 7
TP 6
+5V
R 9 1.5k
REFHACK
CPUCK 88
HACK 88
*DMACK 88
DMACK 88
CLK 8M
CLK 4M

デコード/ROM

DMAC
ADDRESS BUS
DMA HI-ADDRESS
ADSTB
M5 MB74LS245
M17 SN7407
M15 SN74LS670
M6 MB74LS373
M7 MB74LS04
M24 MB74LS08
M23(SC6) IP8237A-5
M16 MB74LS245
DATA BUS

外部インターフェイス

*EB
*QB
RWB
*RAMIHB
*IOIHB
M48 MB487
M59 MB487
C12 0.1μ
C13 0.1μ
+5V
M32 SN74S05
C 10 100p
R 5 100
M38 MB74LS02
M24 MB74LS08
M58 MB74SS08
M47 MB74LS32
M58 MB74S08
M28 MB74LS00
M13 MB74LS20
M57 MB74S10
M3 MB74LS32
MB74LS00
M37 MB74S04
*MEMR
*MEMW
DMAAEN
*DMADL 3CK
*DL 2CK
*IOR
*IOW
*DL 3CK
DL 8CK
*HACK 88
FC 00
*DMAAEN
HACK 88

割り込み信号処理

INT 0
INT 1
INT 2
INT 3
INT 4
INT 5
INT 6
INT 7
NMI
RESETB
M18
MB74S04
M8
MB74LS00
M9
MB74LS10
M18
MB74LS04
R 1
J 2
*IRQ 8
*IRQ1
*FIRQ 1
*IRQ 5
*IRQ 6
*FIRQ 0
*FIRQ 2
*FIRQ 3
*IRQ 0
*IRQ 2
*IRQ 9
INTNDP
*IRQ 4
*IRQ 7
*NMI 0
*NMI 1
*RESET

マザーバス・コネクタ(2/1)

マザーバス・コネクタ(2/2)

CN 1		CN 1	
51	*BUSREQ 0	52	*BUSREQ 1
53	*BUSREQ 2	54	*BUSQEQ 3
55	*BUSACK 0	56	*BUSACK 1
57	*BUSACK 2	58	*BUSACK 3
59	NC	60	*DGRNT 1
63	*IRQ 0	64	*IRQ 1
65	*IRQ 2	66	NC
67	*IRQ 4	68	*IRQ 5
69	*IRQ 6	70	*IRP 7
71	*IRQ 8	72	*IRQ 9
73	*FIRQ 0	74	*FIRQ 1
75	*FIRQ 2	76	*FIRQ 3
77	*NMI 0	78	*NMI 1
79	NC	80	NC
81	NC	82	NC
83	*HALT 88	84	HACK 88
85	NC	86	NC
89	NC	90	NC
91	CPUCK 88	92	DMACK 88
93	NC	94	NC

メイン・インターフェイス
*MR
DATA BUS
*HDDRQR
*HDREGW
*DRVREGW
*DRVREGR
RS0
RS1
*CS
*WE
*RE
MB74S08
MB74S00
MB74S04
MB74LS04
MB74LS138
MB74LS08
MB74S10
MB74LS245
MB74LS157
MB74LS05
DL1
R3 620
R4 620
+5V
IRQ
DRQ
CN9
*RESET
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
RWB
AB0
AB1
AB2
*BUSACKO
*FDCS
*EB
FIRQEN
*IRQ4
*BUSRE

CPU周辺

LOCAL ADDRESS BUS
MAIN ADDRESS BUS
*MREQ
*MREQ
*M1
*RFSH
*RD
M8
MB74LS04
TP8
M9
MB74LS244
M2
MB74LS244
C7 0.1μ
C3 0.1μ
+5V
AB15
AB14
AB13
AB12
AB11
AB10
AB9
AB8
AB7
AB6
AB5
AB4
AB3
AB2
AB1
AB0
ZA15
ZA14
ZA13
ZA12
ZA11
ZA10
ZA 9
ZA 8
ZA 7
M28
MB74S00
M20
MB74S08
3.3k
Z80A(SC1)
MREQ
M1
RFSH
RD
WR
IORQ
NC
VCC
GND
φ
RESET
NMI
INT
BUSRQ
BUSACK
HALT
WAIT
CB1
3.3μF
R1
TP9
R5 330
M3
MB74LS244
C4 0.1μ
DIR
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
Z80φCPU
*RESET
*NMI
*INT
*WAIT
DATA BUS
*RD
WAITZ

システム制御信号

M1
MB487
M14
MB74S10
M28
MB74S00
DL2
J1
M15
SN74S74
M29
SN74S74
M17
SN74S32
M8
MB74LS04
M16
MB74LS74A
M27
MB74LS74A
M20
MB74S08
*WAIT
TP7
RWR
*FB
*QB
*REFCK
MREQ
*RFSH
*RD
Z80φ
*MREQ
*WAITZ
*M1
MRDY

バンク・レジスタ

アドレス・デコード

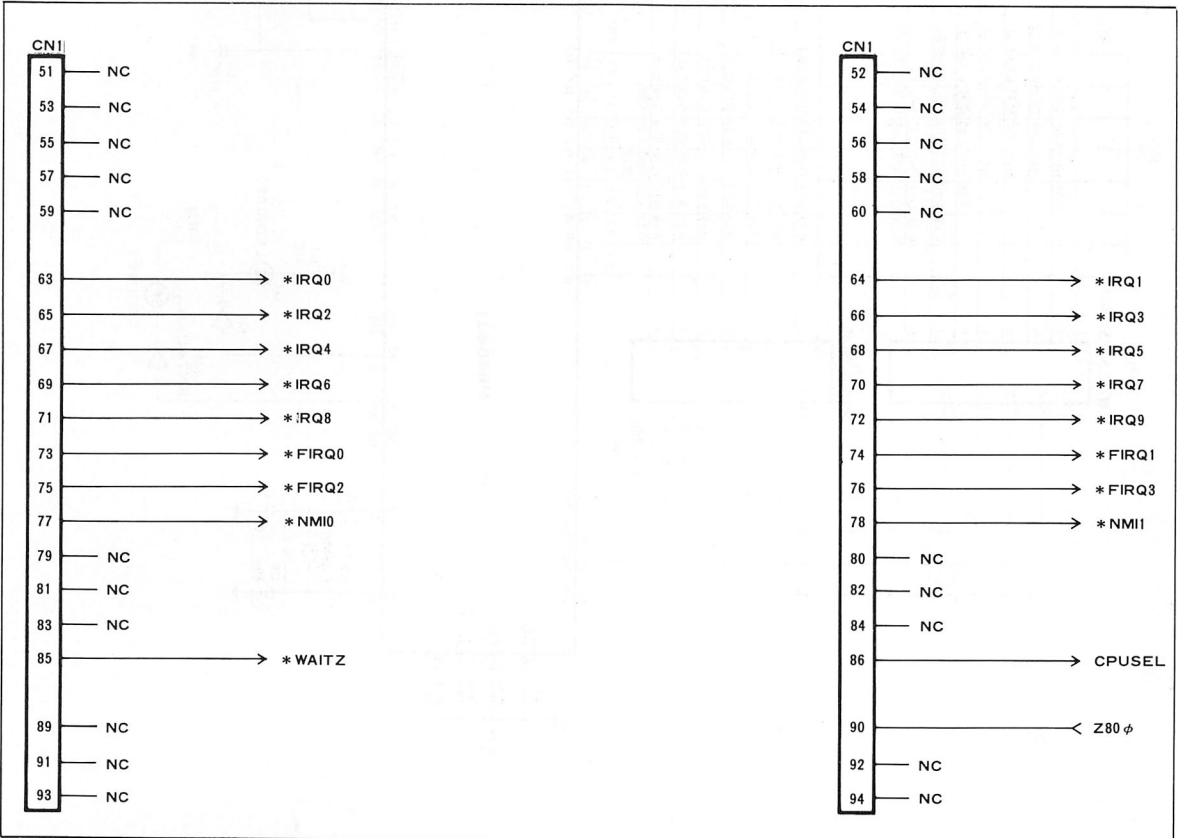
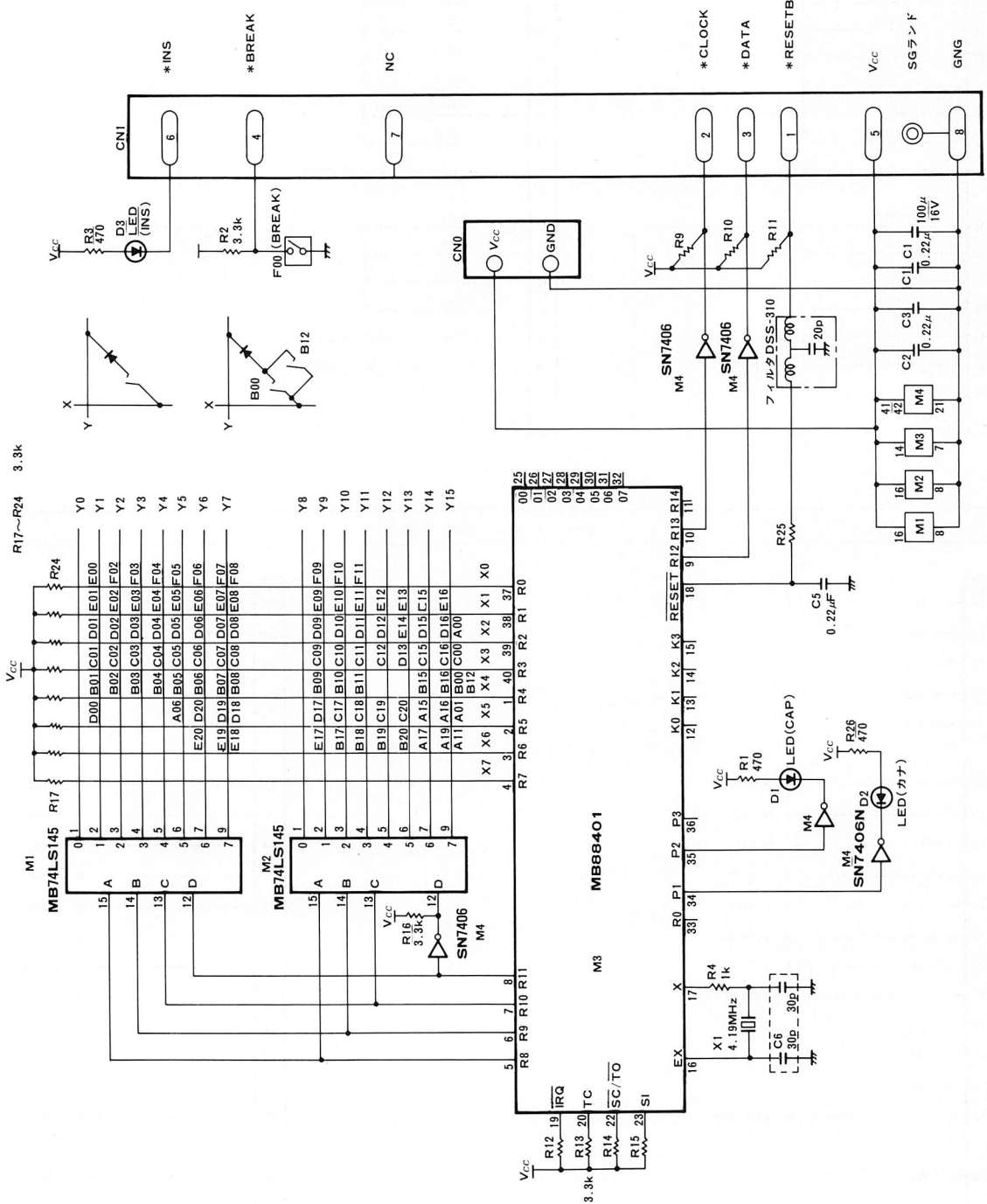
*RD
MAIN ADDRESS BUS
+5V
M18
MB74LS30
M22
MB74LS02
M31
MB74LS32
M32
MB74LS08
M25
MB74LS11
M22
MB74LS02
M21
MB74LS27
M31
MB74LS32
M21
MB74LS27
M23
MB74LS00
M30
MB74LS10
M30
MB74LS10
M30
MB74LS10
M35
MB74LS32
M31
MB74LS32
M26
SN74S32
*MASKSET
M31
MB74LS32
*MASKRD
M35
MB74LS32
M26
MB74LS32
*BANKSET
M35
MB74LS32
M35
MB74LS32
*BANKRD
M8
MB74LS04
WAITZ
*WAITZ
*IOIHB
RWB
M26
MB74LS32
M8
MB74LS04
*EB
DL1
200
J2

割り込み/クロック

マザーバス・コネクタ(1/2)

マザーバス・コネクタ(2/2)

CN1
51 NC
53 NC
55 NC
57 NC
59 NC
63 *IRQ0
65 *IRQ2
67 *IRQ4
69 *IRQ6
71 *IRQ8
73 *FIRQ0
75 *FIRQ2
77 *NMI0
79 NC
81 NC
83 NC
85 *WAITZ
89 NC
91 NC
93 NC

CN1
52 NC
54 NC
56 NC
58 NC
60 NC
64 *IRQ1
66 *IRQ3
68 *IRQ5
70 *IRQ7
72 *IRQ9
74 *FIRQ1
76 *FIRQ3
78 *NMI1
80 NC
82 NC
84 NC
86 CPUSEL
90 Z80 φ
92 NC
94 NC

FM-11実用ソフト紹介

ザ・パソコン給与システム……308
PCA高級財務会計システム　ザ・パソコン会計……310
FM財務……312
利益計画管理システム……314
uniQes-TQC(品質管理)……316
SPERPROシリーズ　販売・仕入・在庫管理……318
DATE ACE……320
漢字dBASE……322
漢字FMCALC……324
JWORD……326
COMAS-WPE1……328
GT-11……330
ターミナル・エミュレーション・プログラム……332

FM-11

ザ・パソコン給与システム

公認会計士 小川哲英氏監修

給与所得の範囲は，俸給，給料，賃金，歳費，年金，恩給，賞与などによる所得をいう．この給与に対し，サラリーマンには源泉所得税がある．これは，給与の形態によって使われる税額表も，月額表，日額表があり，適用する欄はそれぞれ甲欄，乙欄と分かれる．

ついで，健康保険，厚生年金保険などの社会保険料の控除や住民税の控除，さらに組み合い費，財形貯蓄などの天引処理，最後に年末調整処理など幅広く，かつ正確な知識を要求される．

したがって，給与事務は事務経験の豊かなベテランか社会保険労務士に任かせてきたのが，中小企業の現状である．

そこで，いままでのシステムを集大成し，公認会計士　小川哲英氏の監修により完成したのが，『ザ・パソコン給与』である．特徴は以下の通り．

❶見易くわかり易い操作手引書があるので，初心者でもすぐ使える．

❷イージーオーダー方式のシステムになっており，製造業，販売業，サービス業はもとより，病院などの特殊業種にも適応できる．

❸大型コンピュータ並みの豊富な資料が作成できる（給与支給控除一覧表，源泉徴収簿など）．

❹給与，賞与，年調処理だけではなく，人事管理システムとして利用できる．

❺４種類の給与体系（月給，日給月給，日給時給）が同時に処理できる．

❻中小企業向けに，パスワードを付けるなど，労務管理上の配慮をしている．

❼交通費非課税限度額，社会保険料や所得税の計算に用いる定数（料額表，税額表）は，法令の改正などがあった場合，ユーザー自身で容易に変更できる．

システムの概要

❶１枚のデータ・シートで，300人分の給与，賞与（年４回）計算処理ができる．

❷月額，日給月給，日給，時給者の処理が可能．端数調整は金額の指定によって，翌月へ繰越処理ができる．

❸支給項目数は基本給以下計15項目．控除項目４区分まで自由説定で使える．

❹給与・賞与システムの入力データおよび計算結果を利用して，年末調整計算ができる．また，年末調整システムだけを利用するユーザーの便宜のために，単独使用もできる．ただし，月額表乙欄使用者および年収1,000万円以上の人は年末調整の対象外とする．

使いやすくするために

1）手計算で行なっている給与計算が，容易にパソコン給与計算に移行できるように，初心者にもわかり易いマニュアルを用意している．

2）給与事務の初心者でも給与，賞与，年末計算ができるように，個人マスタ登録用紙，給与データ入力用紙，賞与データ入力用紙が用意されている．

3）パソコンでの給与計算は，画面メニューの選択によって行なえるので，パソコンに不慣れな人でも自由自在に行なえる．

4）簡明な操作キーがあるので，便利である．

PF1 次：入力画面などで，必要な入力終了後，次の処理をしたいときに使う．
または，検索データの画面表示で，次の画面を見たいときに使う．

PF2 訂正：データの入力中にデータを訂正したいときにこのキーを使う．プリント中に一旦停止してプリントの印字位置を確認したいときにも使える．

PF3 終わり：このキーを押すと，処理していた業務を強制的に終了し，データの書き込みなど必要な処理を自動的に行なって，メニュー画面に戻る．

PF4 取り消し：入力画面の確認の際，入力データを全部キャンセルする必要がある場合に使う．

⏎：各項目（たとえば基本給，手当など）のデータ入力が終わったとき，および『確認』の画面でＯＫのときに使う．

これだけの操作キーが使えば，『ザ・パソコン給与』が使える．

図１　印字例（一部）49/100

給　与　支　給　控　除　一　覧　表　　58年 11月

氏　名	岩崎　信彦	髙橋　真知子	大藤　幹夫	河野　薫	伊東　雅典	部課合計	総合計
社員コード	0001	0002	0003	0004	0005		
基本給	380,000	150,000	210,000	115,000	168,000	1,023,000	1,023,000
役職手当	250,000	0	0	0	96,800	346,800	346,800
家族手当	0	0	0	0	5,000	5,000	5,000
普通残業手当	0	0	11,000	10,000	0	21,000	21,000
深夜残業手当	0	0	50,000	0	0	50,000	50,000
特別手当	15,000	80,000	150,000	15,000	10,000	270,000	270,000

図2　システムの全体図

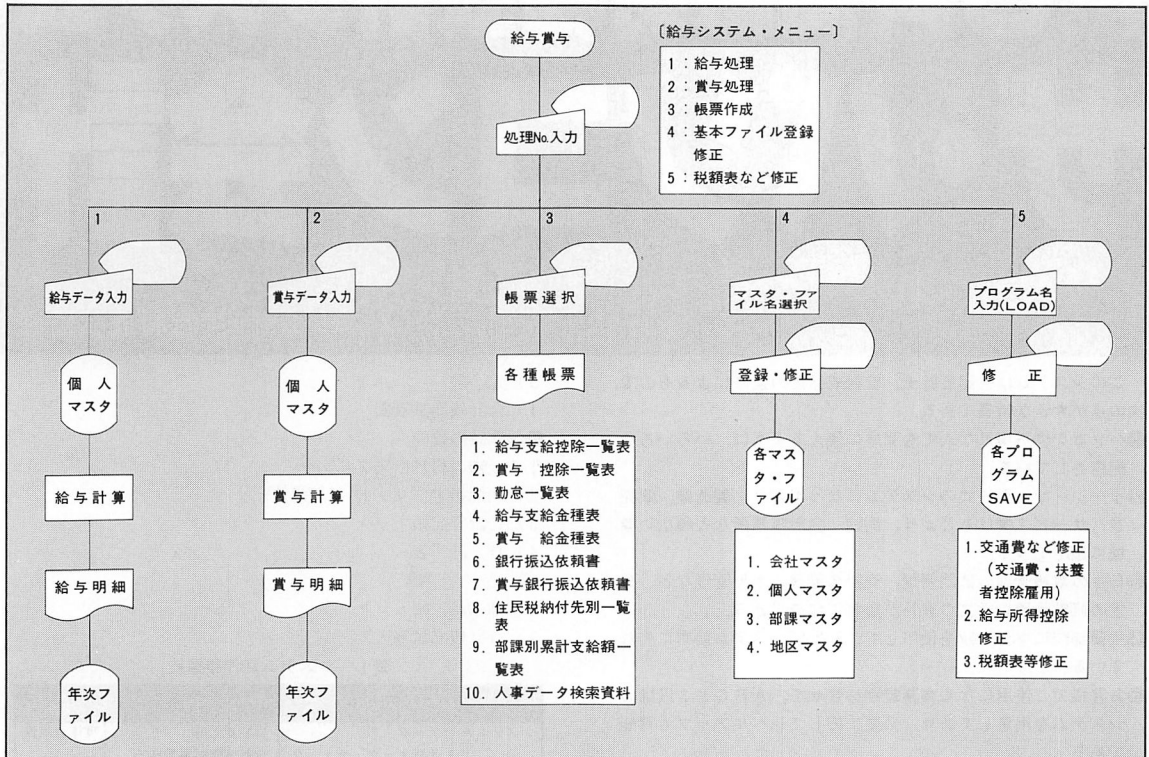



図3　印字例(一部)

PCA給与システム

【給与データ入力】

社員コード　1003　1:氏　名　佐藤　博

項目	値	項目	値	項目	値
2:部課コード	01	16:残業手当(普)		30:財形2控除	
3:有給休暇		17:(深)		31:諸控除	
4:特別休暇		18:基本給	263,000	32:その他控除	
5:生理休暇		19:役職手当	114,250	33:控除	
6:代休暇		20:家族手当	7,000	34:控除	
7:事故欠勤日数		21:技術手当	15,000	35:控除	
8:病欠勤日数		22:諸手当		36:控除	
9:出勤日数(時間)		23:手当		37:控除	
10:遅刻早退回数		24:手当		38:健康保険控除	17,425
11:時間		25:手当		39:厚生年金控除	21,730
12:残業時間平普		26:手当		40:住民税控除	18,270
13:平深		27:課税通勤手当	470	41:有休残日数	12.00
14:休普		28:通勤手当	19,500	42:就労状況	0
15:休深		29:財形1控除	10,000	43:扶養人数	4

経営管理に役立てるために

給与計算と同時に経営管理，人事管理のための資料を作成することができる．

❶部課別支給控除一覧表は見易く，すべての項目につき，部門別に集計できるので，原価管理，予算管理，部門管理を必要としている場合に有効である．

❷労務管理に有用な勤怠一覧表を，個人，部課単位，全社ベースの選択で自由に作成できる．

❸部課別累計支除額一覧表を給与，賞与計算システムのみのユーザーにも作成できるようにしているので，財務会計との照合や，予算作成に有用である．

❹生年月日，入社年月日，職歴などの組み合わせで，自由に入力データを検索することができる．

法的要件を満たすために

1) 所得税源泉徴収簿

国税庁，労働省様式に準拠して，一番面倒な一人別所得税源泉徴収簿を作成することができる．

2) 給与支払報告書

年末調整の結果，給与支払報告書を作成する．

開発会社名：ピーシーエー㈱
販売元：ピーシーエー㈱
問い合わせ先：ピーシーエー㈱
〒東京都渋谷区渋谷3-15-15グリームビル5F
東京☎(03)368-9631(代)
大阪☎(06)315-8637
名古屋☎(052)565-0406
価格：給与・賞与計算システム￥148,000
年末調整システム　￥39,000
使用言語：漢字FBASIC
必要なOS：CP/M86
提供媒体：5インチ・ディスク
マニュアル：p.50 マニュアル別売はなし
サンプル・データ：なし
出荷開始時期：83年11月
最小システム構成：本体FM-11EX（MB25050）
高解像度カラーディスプレイ（MB27311）
ビジネス・プリンタ（MB27402）
追加薄形ミニフロッピィーディスク（MB27609）
漢字ROMカード（MB28111）
拡張RAMカード(128K)（MB22414A）

FM-11

PCA高級財務会計システム ザ・パソコン会計

公認会計士　吉村成弘監修

このシステムは公認会計士，吉村成弘氏の監修によるもので，次の点が大きな特長である．

❶パソコンに不慣れな人でも簡単に使えるように，いろいろな配慮をしている．
❷イージーオーダー式のシステムになっており，製造業，販売業，サービス業はもとより，病院，会計事務所など幅広い業種に適応できる．
❸日々の業績管理，部門管理，売掛金管理，手形管理など，日常の経営管理に役立つ資料が豊富に作成できる．
❹主要部門にマシン語を使用しているなど，システム効率に秀れている．
❺お客様のご使用になる業務範囲に合せて，ＡＢＣと３段階のシステムを用意してあり，必要に応じてレベルアップも可能である．

システムの概要

❶基本設計事項
1）1,000億円未満の会社
2）仕訳数2,000/月
3）１仕訳100億円未満
4）会計処理月などの登録
5）伝票No.の入力有無
6）主勘定科目数240
7）補助科目数　240
8）部門設定　　9
9）摘要コード　200(カナ100)
10）各項目の登録，訂正機能
11）月末，期末自動更新
❷データ入力方法
1）入出金データの自動仕訳
2）単一振替伝票
3）複合振替伝票(N：N)
4）主要48科目ワンタッチ入力
5）常時１件前の入力データ表示
6）データの訂正，取り消し
❸データの検索(仕訳データのチェック用)
1）伝票番号
2）日　付
3）勘定科目
4）金　額
5）検索データの訂正，取り消し
❹画面表示
1）漢字による操作指示
2）検索データ
3）元　帳
4）合計残高試算表
❺帳票時の機能
1）元帳の科目指定プリント
2）期間指定プリント(日付単位)
3）仕訳日記帳
4）日　計　表
5）元　　帳
6）補助元帳
7）合計残高試算表

表１　システム別作業機能

	日常	月次	決算
システムＡ	データ入力 仕訳データ・チェック 検索，訂正，取り消し	総勘定元帳 合計残高試算表B/S 合計残高試算表P/L	賃借対照表
システムＢ	要約日計表	補助元帳 補助科目残高一覧表 手形管理表 取り引先別売り掛け，買い掛け	
システムＣ		資金繰り実績表 経営分析表 部門別損益計算書 部門別P/L一覧表 月次残高推移表 取り引き先別総合収支明細書	三期P/L比較 棒グラフ

使いやすくするために

❶現状の伝票類がそのまま使える
　現状の伝票や出納帳が，そのままでパソコンとソフトを買えば，その日から会計処理が合理化される．
❷ワンタッチ・キー
　主要48科目が科目名をキーに貼って使えるようになっているので，日常よく使う科目は，コードを使わなくても，ワンタッチで科目の入力ができる．
❸簿記の知識があまりなくてもOK
　科目残高，登録とか仕訳データの入力など，会計処理特有の借方，貸方という知識がなくても使える．
- 残高の登録は，所定の科目に残高を入力すれば，借方，貸方はパソコンが管理している．また，お客様が使用される科目は，その科目に金額が入っているかどうかで，パソコンが自動的に判断して，プリンタには金額の入っている科目のみを印字するようになっている．
- データの入力も入出自動仕訳，伝票イメージ入力ができる．
- 各種資料の作戦は，『MENU2』『MENU3』で作りたい資料

表1(一部分)

＊　仕　訳　チェック　リスト　＊

会社名　PCAサンプル　　頁 1

NO.	年月日	コード	借方 科目	(補助科目)	コード	貸方 科目	(補助科目)	金額	摘要
2	58.08.01	111	現金		511	売上高 1		148,000	高級財務会計システム
3	58.08.01	151	受取手形		000	諸口		1,000,000	
3	58.08.01	121	当座預金 1		000	諸口		476,000	
3	58.08.01	741	支払手数料		000	諸口		4,000	
3	58.08.01	000	諸口		152	売掛金 1		1,480,000	
4	58.08.01	111	現金		511	売上高 1		30,000	減価償却計算システム
5	58.08.01	111	現金		512	売上高 2		746,500	FM-7 システム
6	58.08.01	111	現金		152	売掛金 1		500,000	山本製作所
7	58.08.01	733	交際費		111	現金		12,500	篠木商店 接待
8	58.08.01	732	広告宣伝費		111	現金		200,000	パンフレット印刷
9	58.08.01	726	旅費交通費		111	現金		1,200	鈴木 タクシー代
10	58.08.01	121	当座預金 1		111	現金		500,000	.
11	58.08.01	152	売掛金 1		511	売上高 1		1,480,000	FM-11システム，高橋商事
12	58.08.01	615	仕入高 1		312	買掛金 1		500,000	Aプリンター，富士通
13	58.08.01	151	受取手形		000	諸口		1,000,000	FM-11システム，KK舘田
13	58.08.01	121	当座預金 1		000	諸口		476,000	FM-11システム，KK舘田
13	58.08.01	741	支払手数料		000	諸口		4,000	FM-11システム，KK舘田
13	58.08.01	000	諸口		511	売上高 1		1,480,000	FM-11システム，KK舘田
14	58.08.01	111	現金		511	売上高 1		235,600	高級財務会計システム
15	58.08.01	121	当座預金 1		511	売上高 1		3,560,000	JB3000システム*2
16	58.08.03	615	仕入高 1		311	支払手形		3,650,000	IBMシステム，10.03
17	58.08.04	741	支払手数料		111	現金		3,600	ABCカード
18	58.08.06	752	保険料		131	普通預金 1		36,900	8月分
19	58.08.09	111	現金		511	売上高 1		36,500	簡易財務システム
20	58.08.10	151	受取手形		511	売上高 1		3,650,000	PC ＊2台
21	58.08.11	615	仕入高 1		311	支払手形		2,560,000	MZセット， 11.13
22	58.08.15	751	賃借料		131	普通預金 1		250,000	8月分家賃
23	58.08.01	121	当座預金 1	101 A銀行当座	511	売上高 1	101 一般売上	1,365,800	FM-8システム
24	58.08.01	121	当座預金 1	102 B銀行当座	511	売上高 1	103 特別売上	1,265,000	PC9800システム
25	58.08.01	121	当座預金 1	103 C銀行当座	511	売上高 1	102 業者売上	5,600,000	HC-20 20台
26	58.08.01	131	普通預金 1	101 A銀行普通	511	売上高 1	103 特別売上	4,020,000	高級財務システム 30セット
27	58.08.01	131	普通預金 1	102 B銀行普通	511	売上高 1	102 業者売上	1,365,000	高級財務システム 7セット
28	58.08.01	131	普通預金 1	103 C銀行普通	511	売上高 1	102 業者売上	2,650,300	簡易財務システム 30セット
29	58.08.15	615	仕入高 1		311	支払手形		3,560,000	AC－9　4台，11.15
30	58.08.15	615	仕入高 1		312	買掛金 1		4,562,000	プリンター20台， 藤波商店
31	58.08.01	615	仕入高 1		312	買掛金 1		4,560,000	ディスプレイ30台，長州商事
32	58.08.01	122	当座預金 2		511	売上高 1		1,265,000	FM－8システム
33	58.08.31	131	普通預金 1		511	売上高 1		265,000	ディスケットドライブB
34	58.08.01	132	普通預金 2		511	売上高 1		20,600	フロッピー20枚
35	58.08.01	141	通知預金		122	当座預金 2		1,000,000	.
36	58.08.01	141	通知預金		131	普通預金 1		1,200,000	.
37	58.08.01	142	その他流動預金		122	当座預金 2		500,000	.
38	58.08.01	111	現金		143	定期預金 1		500,000	
39	58.08.01	121	当座預金 1		144	定期預金 2		560,000	
45	58.08.01	121	当座預金 1		151	受取手形		1,000,000	浜口商店決済
46	58.08.01	152	売掛金 1		511	売上高 1	102 業者売上	360,000	Aプリンター
48	58.08.01	615	仕入高 1		311	支払手形	101 A銀行 1月	1,360,000	仁科商事，，11.01
49	58.08.01	311	支払手形	102 A銀行 2月	121	当座預金 1	102 B銀行当座	2,300,000	テスト 1
50	58.08.01	321	短期借入金		122	当座預金 2		250,000	テスト 2
51	58.08.01	322	未払金		122	当座預金 2		325,000	テスト 3
52	58.08.01	616	仕入高 2		311	支払手形	101 A銀行 1月	2,365,000	テスト 4

表2(一部分)

＊　日　計　表　＊

昭和 58 年 8 月 31 日

科目名	コード	前月残高	借方合計	貸方合計	当月残高
1 現金	111	517,102	0	0	517,102
2 当座預金 1	121	12,625,700	0	0	12,625,700
3 普通預金 1	131	12,847,920	265,000	0	13,112,920
4 通知預金	141	6,700,000	0	0	6,700,000
5 定期預金 1	143	8,000,000	0	0	8,000,000
6 その他固定預金	149	300,000	0	0	300,000
7 ***	XXX	0	0	0	0
8 ***	XXX	0	0	0	0
9 ***	XXX	0	0	0	0
10 ***	XXX	0	0	0	0
11 ***	XXX	0	0	0	0
12 ***	XXX	0	0	0	0
111-149 計		122,578,253	265,000	0	122,843,253
13 受取手形	151	60,596,500	0	0	60,596,500
14 売掛金 1	152	25,809,000	0	0	25,809,000
15 貸倒引当金	159	-563,900	0	0	-563,900
151-159 計		93,624,710	0	0	93,624,710
16 商品 1	161	4,568,900	0	0	4,568,900
17 製品	165	1,659,800	0	0	1,659,800
18 原材料	166	656,900	0	0	656,900
資産合計		431,991,546	265,000	0	432,256,546
19 支払手形	311	38,633,755	0	0	38,633,755
20 買掛金 1	312	893,170	0	0	893,170
21 ***	XXX	0	0	0	0
311-316 計		51,601,825	0	0	51,601,825
22 短期借入金	321	1,000,000	0	0	1,000,000
23 未払金	322	0	0	0	0
24 預り金	326	528,750	0	0	528,750
25 ***	XXX	0	0	0	0

番号を選べば，後はパソコンが自動的に分類，集計処理をして，指定した資料を作成してくれる．

❹パソコンに不慣れな人でも自由自在

基本的に次の4つの操作キーと画面対話式で，パソコンに不慣れな人でも自由自在に使える．

PF1(次)・次の項目，または次の処理をする．
・検索データの画面表示で，次の画面を見る．

PF2(訂正)・データの入力中に，データの訂正をする．
・プリントの印字位置を確認したいときに使う．

PF3終わり・処理していた業務が終了したことを意味し(画面上の処理が終わったときに押す)，データの書き込みなど必要な処理を自動的に行なってメニュー画面に戻る．

注)画面上の『確認』のときに押すと画面上のデータは無効になる．

・各項目(たとえば日付，科目など)のデータ入力が終わったとき，および『確認』の画面でＯＫのときに使う．

たったこれだけの操作キーが使えれば，財務会計が処理できる．

❺資料の作成はメニューから番号を選ぶだけ

いろいろな管理資料が作成できるように用意している．これら資料の作成はメニューから必要な資料の番号を選べば良い．

❻スタンド式カラー操作説明書

説明書は図解入りカラー説明で，しかもスタンド式になっているので，CRT(テレビ)の上に立て，必要な部分を見れば良いようになっている．

❼その他

ディスケットに記録されたデータの『訂正』，『取り消し』など，通常の事務処理に必要な処理が，簡単にできるようになっている．

正確かつ迅速な処理のために

❶プログラムやデータの保護

操作ミスなどで，プログラムかデータを壊したら大変である．使いやすいということは，それだけでかなりの安全性があることを意味する．プログラムやデータ保護のために，万一を考えた配慮もいろいろと組み込んでいる．

❷入力処理が速い

慣れればそれだけ作業能率が向上する．

❸主要部分はマシン語

1,000件のデータならわずか１～２分で処理する．

広範囲に適応するために

❶科目名の変更が自由

主勘定科目名はあらかじめ，製造販売業用に登録されているが，科目名を変更することによりいろいろな業種に活用できる．

たとえば次のように変更する．

(ＰＣＡ登録)	(医師業)	(建設業)
511 売上高１	⇒保険窓口収入	完成工事高
512 売上高２	⇒保険請求収入	
513 売上高３	⇒自由診療収入	兼業事業売上高
⋮ ⋮		
〃 7		
625 売上原価	⇒医業原価	完成工事高

とすればよい．

❷使わない科目はプリントしない

主勘定は240種類が用意されているが，使わない科目はプリントしない．

開発会社名：ピーシーエー㈱
販売元：ピーシーエー㈱
問い合せ先：ピーシーエー㈱
〒150 東京都渋谷区渋谷3-15-15グリームビル5F
東京☎(03)368-9631(代)
大阪☎(06)315-8637
名古屋☎(052)565-0406
価格：システムＡ(基本) ￥98,000
システムＢ(Ａ＋補助，他) ￥148,000
システムＣ(Ｂ＋部門，他) ￥198,000
使用言語：漢字BASIC，マシン語
必要なOS：CP/M(カナは不要)
提供媒体：５インチ・ディスク
マニュアル：あり，p.68(図解入りカラー，スタンド式)
マニュアル別売：なし．ただし第一法規出版の『ザ・パソコン会計』￥1,600で詳しく解説．
サンプル・データ：なし．
出荷開始時期：カナ版 '83年２月
漢字版 '83年11月

FM-11

FM-ZAIMU

財務管理システム

FM-ZAIMUは仕訳伝票の入力により，集計計算を行ない，仕訳日記帳から決算報告書，経営管理帳表の出力ができる本格的財務管理システムである．また，日本語ワード・プロセッサの機能を組み込むことにより，漢字の取り扱いが簡単になり，会計専用機なみの機能を低コストで実現している．

❶経理業務の時間削減
❷経理情報のタイムリーな供給
❸経理情報の多角的利用

勘定科目の入力

仕訳伝票を入力するときの科目入力は，従来の4桁の科目コード入力の他に，頭文字入力，画面入力，および固定仕訳入力が行なえる．

頭文字入力は，科目名を簡略した3文字以内の頭文字を入力する方法で，科目コードがわからなくても，科目名を連想することで入力できる．また，頭文字はユーザが自由に設定することができる．

画面入力は画面下に科目名の一覧を表示して，その中から番号で選択させる方法である．画面入力では，科目コードや頭文字がわからない場合でも，科目一覧表を見えなくても，その場で勘定科目を入力できる．

固定仕訳入力は発生頻度の高い，複合仕訳のパターンをあらかじめ登録しておき，科目入力時にＰＦキーで取り出す方法である。固定仕訳は最大9種類まで登録することができ，PF1 から PF9 までが割り当てられている．

この入力方法を使うと，勘定科目が自動的に入力され，金額だけ入力することになり，入力効率を上げることができる．

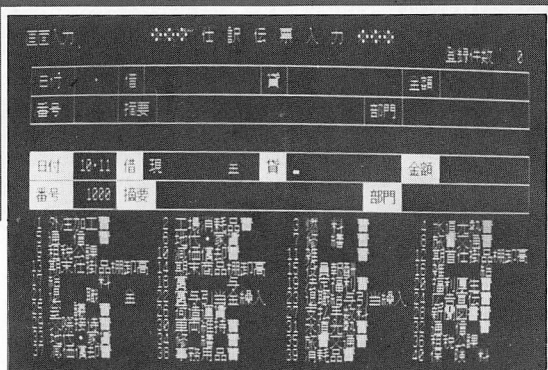
このように科目入力は，従来のコード入力のみに比べて，覚えやすく簡単に入力することができる．

カナ漢字変換

仕訳伝票の摘要入力，科目名の修正など漢字入力は，すべてカナ-漢字変換方式で入力することができる．入力された漢字はそのまま帳表に出力することができ，漢字の取り扱いが非常に簡単になった．

帳表出力

必要な帳表が，その会社のフロッピィーディスクをセットすることにより，いつでも出力することができる．たとえば，月中でも出力すれば，その時点の残高が出力される．

また，科目名や摘要が漢字で，白紙のストック・フォームに印刷するので，見やすく指定用紙（印字済用紙）を用意する必要がない．

操作性と柔軟性

オペレーション指示メッセージとファンクション・キーの表示や，エラーチェック機能の強化など，初めて扱う方でも安心して操作することができる．

また，勘定科目の変更が自由にでき，いろいろな業種の企業や法人でも，運用することができる．さらに，更新後に月を遡って，仕訳伝票を修正することが可能になっており，柔軟性のあるシステムを実現している．

経営管理

オプションにより，経営の総合的な判断を行なうための帳表やグラフを作成することができる．また，部門別管理を行なうことにより，部門ごとの経費，予算を把握することができる．

高速処理

伝票の分類処理など，スピードを要求されるところは，マシン語を使い，高速処理を可能にした．

操作手順

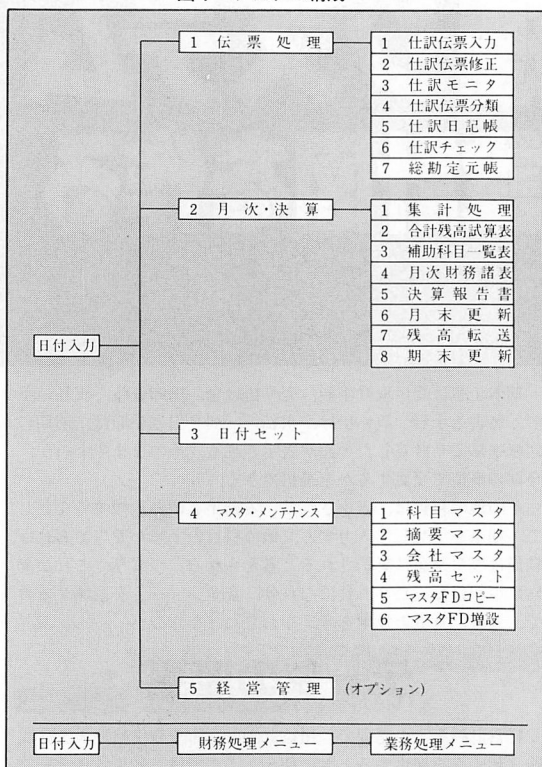
FM-ZAIMUは図1のようなシステム構成になっている．処理メニューが2段階に分かれ，各処理メニューで処理番号を入力すると，該当する次の処理に進む．

FM-ZAIMUのFM-11システムは，オート・スタート機構になっている．したがって，フロッピィーディスクをセットして電源を投入すると，FM-ZAIMUが自動的に起動し日付入力画面が表示される．

ここで会社名を確認して，6桁の日付を入力すると，第1段階の財務処理メニューが表示される．

たとえば，仕訳伝票入力を行なう場合は，処理番号『1』を入力して，伝票処理を選択する．次に伝票処理メニュー画面が表示されるので，再度処理番号『1』を入力すると，仕訳伝票入力処理が起動される．

仕訳伝票の入力形式は，単一仕訳入力と複合仕訳入力があり，最初は単一仕訳入力となっており，ＰＦキーで切り換えること

図1　システム構成

ができる.

入力項目は日付，伝票番号，借方科目，貸方科目，金額および摘要で，部門コードは必要に応じて入力する．複合仕訳は，貸借それぞれ7明細まで入力できる．

仕訳伝票の入力が終わると，集計処理により，純利益が自動的に算出されて，財務諸表を出力することができる．伝票入力後の操作は主に各種帳表の出力指示となり，処理を選択して出力される．

導中効果

FM-ZAIMUを導入して，パソコンで経理業務を行なうと次のようなメリットが得られる．

経理業務の時間削減

仕訳伝票を入力するだけで，自動的に転記集計が行なわれ，財務諸表，補助簿など帳表が作成できるため，いままで帳簿への記入，試算表の作成などにかかっていた時間が大幅に削減できる．

特に決算業務は決算修正伝票の入力により，即時に決算報告書を得ることができる．

経理情報のタイムリーな供給

経理情報の価値は，その正確性とタイムリー性にある．伝票入力時のデータ・チェックがされていれば，転記ミス，計算ミスは起こらないので，正確性は保証される．また，伝票入力時にデータ・ファイル（帳簿）が更新されているので，いつでも最新の経理情報を得ることができる．

経理情報の多角的利用

経営の収益性，流動性，生産性の分析を行なった経営分析表，時系列比較を行なった前期比較表，三期比較表，予算と実績を比較した予算管理表など，豊富な経理情報を得ることができる．また，総勘定元帳や補助科目一覧表を利用することにより，売掛，売上，仕入などの管理を行なうこともできる．総勘定元帳は補助科目を指定すると，補助簿として使うこともできる．

財務仕様

	FM-11
科目入力	コード入力 頭文字入力 画面入力 固定仕訳入力 （オプション
科目数	320
補助科目	任意に設定
金　額	出力帳表　11桁(一千億未満) 伝票入力　10桁(百億未満)
決　算	1年決算
摘　要	漢字14文字
摘要入力	固定摘要のコード入力 カナ漢字変換入力
固定摘要数	126件
伝票枚数	3,800枚/月
出力帳票 (標準システム)	仕訳モニタ 仕訳日記帳 仕訳チェック・リスト 総勘定元帳(補助元帳を含む) 補助科目一覧表 月次財務諸表 (月次貸借対照表など4帳表) 決算報告書 (貸借対照表など4帳表)
(オプション)	経営分析表 要約比較表 予算管理表 前期比較表 三期比較表 月次推移表 科目別推移図 三科目比較推移図 資金線実績表 部門別損益計算書

開発会社名：三和ビジネス㈱
販　売　元：三和ビジネス㈱
問い合せ先：三和ビジネス㈱
〒神奈川県横浜市中山区下町75荻野ビル８Ｆ
☎(045)641-6401(代)
価格：標準システム　￥150,000
　　　オプション　　￥50,000
使用言語：F-BASIC，アセンブラ
必要なOS：なし(F-BASIC下で稼動)
提供媒体：５インチまたは標票ディスク
マニュアル：約p.90
マニュアル別売：なし（FM-ZAIMUをトラブルシュートやQ&Aを取り入れて，わかりやすく解説した『会計ワープロ入門』を出版している)
サンプルデータ：勘定科目はあらかじめ標準的なものを200科目設定してある．
出荷開始時期：FM-11 '83年８月
最小システム構成：FM-11（漢字ROMカード含む)
　グリーンCRTディスプレイ
　フロッピィーディスク・ドライブ(2ドライブ)
　シリアル・ドット・プリンタ

FM-11

利益計画管理システム 目標利益達成の設計図を作る

経営計画という言葉は，いろいろなところで使われ，それによって意味を異にしていることがある．開発担当の私たちは，経営計画を『目標利益達成の設計図』と考えている．

このため，目標利益額の決定には，経営者の戦略的見地からの必要額と，生産，販売の現場からの達成額とのギャップを検討することが欠かせない．

そして，企業にはいくつかの事業部，商品群，単一商品に頼る場合でも，変動比率の異なるいくつの流通経路を持っている．その占める割合が変われば，総売上高が同じでも限界利益の額は異なってくる．

目標利益額決定のための販売部門の達成可能売上高は，限界利益率の異なる，セグメント別のものでなければならない．こうした事業部，商品群，流通経路などを，ここではプロフィット・センターと呼ぶ．

経営計画を策定した後は，実績との対比をしなくては意味がなく，設計図通りかどうかをチェックしてこそ活かされる．

この利益計画管理システは，計算，作表をコンピュータが，策定を人間が担当する．ここで，このシステムの特徴を示す．

❶考える部分を人間が，単純作業をコンピュータが担当するので，このシステムだけに必要なデータはない．
❷予実比較表が作成できるので，差異分析が容易．
❸期末予想損益計画書は予算と実績を結びつけるので，対応策の必要性が早期につかめる．
❹期末予想貸借対照表は現金の当期残高により，資金調達，運用の問題点を明らかにし，増減額でその問題点がどこにあるのかを明白にする．
❺パッケージ化してあるので，企業，コンサルタント業にも利用できる．

システムの概要

システムをパッケージ化するために，会社名，決算月，金額単位（千，万，10万単位の3種類から選択），プロフィット・センターの数（15まで），名称，固定費項目数(15まで)，固定費項目名，実績変動費の入力方法（プロフィット・センター別か，一括か）を基礎データとして，登録しておく．この後，種々のデータを入力しておく．

期末予想損益計算書

実績の累計と残りの期間の予算合計により，期末予想損益計算を行なう．このため，実績の登録はすべて同じ月までとする．もし，実績が登録される前なら，予算だけで作成する．

期末予想貸借対応表

期末予想は受け取り手形，売り掛け金，棚卸資産，支払い手形，裏書き手形，買い掛け，未払い，割り引き手形は，前期の回転率により計算した金額が表示される．そのままでよいか，今期の事情で変更するかを選択できる．

その他の科目は，表示されている前年実績と比較するなどして，期末を予想して入力する．他の科目の予想が妥当であれば，結局は現金預金に集約されると考えられる．つまり，これが異常に少額になったり，負になれば，資金ショートを意味するので，資産，負債の運用を考えなおさねばならない．

データの変更

経営計画策定の過程で，損益計画計算書や貸借対照表を作成後，データを変更するときに使う．ここでは，変更するデータの区分を選択する．変更後にシミュレーションを選択すると，新しいデータによる損益計算書や貸借対照表が得られる．

経営計画の策定についての考え方

私たちの考えている経営計画の策定について，最後に述べる．

経営計画という言葉は，いろいろなところで使われ，それによって意味を異にしている．私たちは経営計画を『目標利益達成の設計図』と考えている．企業は営利を目的としているから，目標利益は大きければ大きいほどよいが，短期的には生産力，販売力に企業の能力の限界がある．そのために，目標利益額の決定には，経営者の戦略的見地からの必要額と生産，販売の現場からの達成可能額とのギャップを検討することが欠かせない．

一般的にみて，企業と市場との関係は，買手市場であるといえるので，具体的な検討はまず販売の達成可能売上高から，変動費を差し引いた限界利益率で予完固定費＋必要利益を除いて損益分岐点を求めて行なうことになる．

総売上高のうちに，変動比率の異なる流通経路の占める割り合いが変われば，総売上高が同じでも限界利益の額は異なる．だから過去の損益計画書から算出された，全社平均の限界利益率を使って目標利益を計算することは意味がなく，目標利益額決定のための販売部門の達成可能売上高は，こうした限界利益率の異なるセグメント別のものでなければならない．事業部や商品群，流通経路などを私たちはプロフィットセンターと名づける．

目標利益額決定の資料として提出される販売データは，直ちに満足のいく目標利益額を築くものでないことが多く，不足額をクリアするために，プロフィットセンター別に詳細な検討に入り，ここで不足部分をクリアするための方策が決定される．私たちは経営計画策定のなかで，この不足部分をどのような方策で解消してゆくかの検討過程が人材を育て，市場との連帯を

図1

```
**   キソデ゛ーターノ INPUT   **

1 カイシャメイ(20シ゛イナイ)

2 ケッサンツ゛キ
                    月

3 キンガ゛クタンイ (1)センエン (2)マンエン (3)10マンエン

4 プ゜ロフィットセンターノ カズ゛ (1)-(15)

5 コテイヒノ ヒモクノ カズ゛ (1)-(15)

6 ヘント゛ウヒシ゛ッセキ INPUTノ シカタ
  (1)プ゜ロフィットセンターベ゛ツ (2)イッカツ
```

プロフィットセンターの名称や固定費の費目は次の画面で入力します

図2

```
* シサンノブ゛ノ INPUT            キンガ゛クタンイ:センエン
      カ モ ク        キ ン ガ゛ ク

1  ゲ゛ンキンヨキン
2  ウケトリテガ゛タ
3  ウリカケキン
4  タナオロシシサン
5  ソノタリュウト゛ウシサン
6  トリタテフノウミコミ
7  ショウキャクシサン
8  ソノタユウケイコテイ
9  ムケイコテイシサン
10 トウシトウ

      シ サ ン ケ イ
```

図3

```
* フサイノブ゛・シホンノブ゛ノ INPUT    キンガ゛クタンイ:センエン
      カ モ ク        キ ン ガ゛ ク

1  シハライテガ゛タ
2  ウラガ゛キテガ゛タ
3  カイカケ・ミハライ
4  カリイレキン
5  ワリビ゛キテガ゛タ
6  ショウヨヒキアテキン
7  タイショクヒキアテキン
8  ソノタノフサイ
    ( フサイケイ )
9  シホンキン
10 ホウテイシ゛ュンビ゛キン
11 ツミタテキントウ
12 トウキリエキ
    ( シホンケイ )

    フサイ・シホンケイ
```

図4

```
        ** キマツヨソウP/L **          タンイ:センエン
      ADVENTURE COMPANY
                     ヨ サ ン     シ゛ッセキ     ゴ゛ウケイ
     コ ウ モ ク    7月- 2月     3月- 6月

ウリアゲ゛ダ゛カ
ヘント゛ウヒ

 ゲ゛ンカイリエキ
コ テ イ ヒ

 エイギ゛ョウリエキ
エイガ゛イシュウエキ
エイガ゛イヒヨウ

 トウキリエキ
```

図5

```
              ** キ マ ツ ヨ ソ ウ B/S **
      ADVENTURE COMPANY                          タンイ:センエン
                  シ サ ン ノ ブ゛                          フサイ・シホンノブ゛
  コ ウ モ ク   ト ウ キ   ゼ゛ ン キ   ゾ゛ウゲ゛ン   コ ウ モ ク   ト ウ キ   ゼ゛ ン キ   ゾ゛ウゲ゛ン

ゲ゛ンキンヨキン                         シハライテガ゛タ
ウケトリテガ゛タ                        ウラガ゛キテガ゛タ
ウリカケキン                            カイカケ・ミハライ
タナオロシシサン                        カリイレキン
ソノタリュウト゛ウ                      ワリビ゛キテガ゛タ
トリタテフノウミコミ                    ショウヨヒキアテキン
ショウキャクシサン                      タイショクヒキアテ
ソノタユウケイコテイ                    ソノタノフサイ
ムケイコテイシサン                      シホンキン
トウシトウ                              ホウテイシ゛ュンビ゛
                                        ツミタテキントウ
                                        トウキリエキ

  シサンケイ                              フサイ・シホンケイ
```

強くする意味で，最も大切な部分だと考えている．

経営計画は，このほかに固定費内容の検討，営業外損益の検討を加えてできあがるが，過程では各プロフィットセンターの売上高，変動費，固定費が変われば，どう変わるのかなどを幾通りも計算してみる必要がある．

また，経営計画は期初に策定し，あとは実績との対比をしないのでは意味がなく，経営計画を『目標利益達成の設計図』とすれば，設計図通り行われているか，設計変更は必要かどうかを，チェックしてこそ経営計画はいかされると思う．

私たちの経営計画シミュレーション・システムは，経営計画の策定の過程では，考える部分は人間が担当し，面倒な計算，作表をコンピュータが担当する．また，実績対比において，この実績で以後計画通りならば期末はどうなるかの計算や，予実比較表の作成をコンピュータが担当するとの考え方で作成している．

開発会社名：ピーシーエー㈱ 大阪
販　売　元：ピーシーエー㈱
問い合せ先：ピーシーエー㈱
大阪　☎ (06)315-8637
東京　☎ (03)368-9631
名古屋　☎ (052)565-0406
価格：￥98,000
使用言語：F-BASIC
必要なOS：なし
提供媒体：５インチ・ディスク
マニュアル：p.60
マニュアル別売：利用解説書出版を予定．
サンプル・データ：あり
出荷開始時期：59年１月
最小システム構成：本体　FM-11EX.
CRT.　グリーン
シリアルドット・プリンタ

FM-11

ユニークス uniQues-TQC 品質管理プログラム

よい商品，よい製品をユーザーに提供するために，検査作業を工程を組み入れ，厳重な検査が必要になる．そこでは，個々の検査結果で製品の良否を判定するだけでなく，製品グループ全体，会社全体，あるいは一定期間の検査結果を統計学的品質管理手法を使って，一般的には管理されている．

この統計学的品質管理手法を用いたプログラム・パッケージがuniQues(ユニークス)-TQCである．このソフトで次の管理図などを作ることができる．

❶$\bar{x}$-R管理図
❷P管理図
❸Pn管理図
❹C管理図
❺U管理図
❻ヒストグラム(生データ，度数データ入力)
❼パレート図
❽計算，グラフが自動作成
❾操作指示は漢字表示
❿グラフ表示は，画面プリンタの両方が可能
⓫データ蓄積可能

システム概要

uniQues-TQCは以下の業務で構成される．

I ①データ入力(計測データでも度数データでも可)
②ヒストグラム作成
II 管理図作成(任意の管理図発行)
III コピー処理

操作方法はすべて，コマンド・メニュー方式を採用し，しかも漢字表示で見やすい．ここで，行ないたい仕事は，表示に従って，番号を入力すればよい．

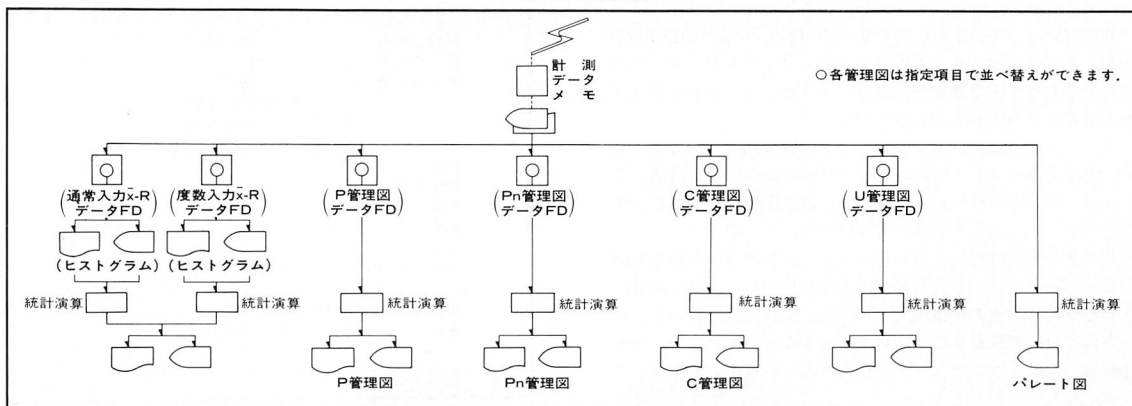
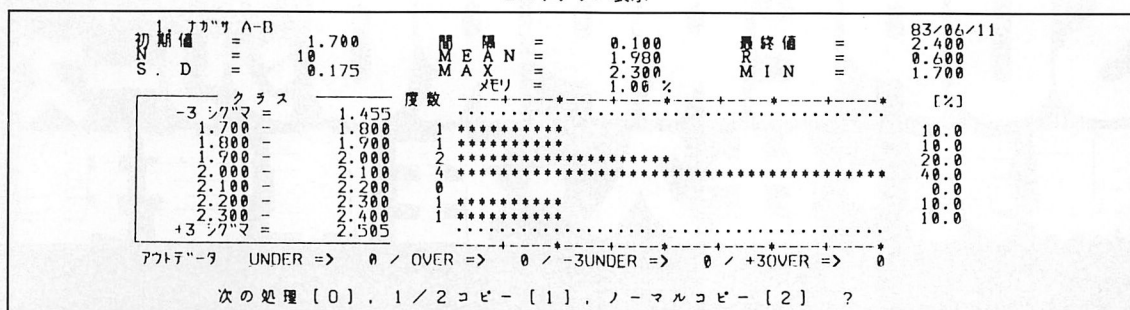
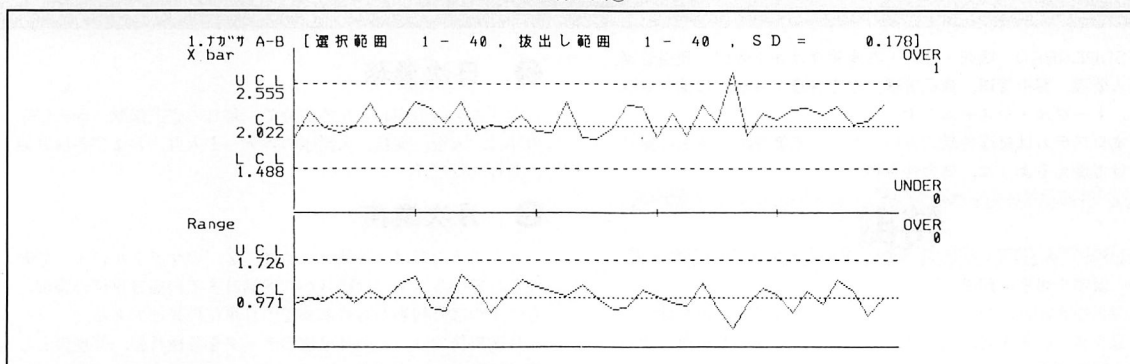
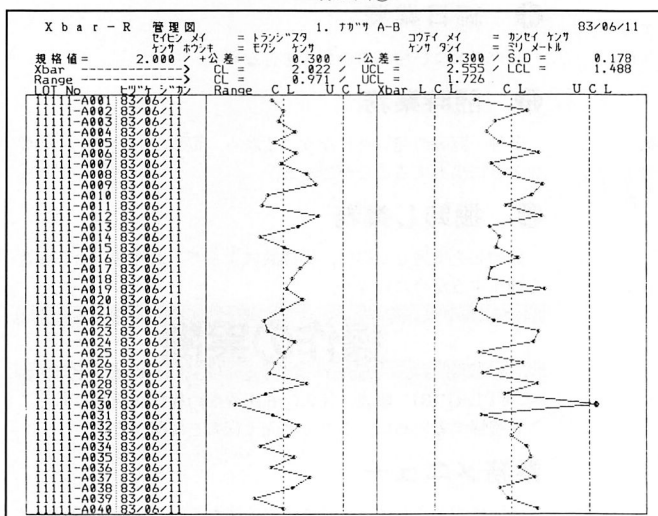
また，誤ったデータ・ディスケットを入れても，正しいディスケットが入るまで，入れ換えるよう指示するようになっているので，安心して使える．

ただし，不意の停電，断線などがあると，データが破壊される可能性があるので，予備を必ず作成しておくべきである．

計測処理件数は，5インチ，8インチともに，計測項目30，計測値生データ400件であるが，計測値は計測項目ごとに，8インチは730件，5インチは220件となっている．

作成管理図とその利用目的

$\bar{x}$-R 管理図	①$\bar{x}$管理図　分布の平均値の変化を見る図 ②R管理図　分布の幅バラツキ（範囲）の変化を見る図
P 管理図 Pn 管理図	製品やサンプルの中に，不良品がどれだけ存在するかという特性を見る図
C 管理図 U 管理図	ある一つの製品のなかに欠点（defects）が何個あるかを見る図
ヒストグラム	①入力された計測データがどんな分布状態をしているか，どんな値を中心にバラツキをもっているか，分布が統計的にどのような型をしているかを見る図　②ヒストグラムは，$\bar{x}$-R管理図の計測データ入力終了後作成
パレート図	不良項目の累積曲線を描いて，原因別，状況別の問題点を見る図

システム作業概略

ヒストグラム表示

$\overline{x}$-R管理図①

$\overline{x}$-R管理図②

総合メニュー

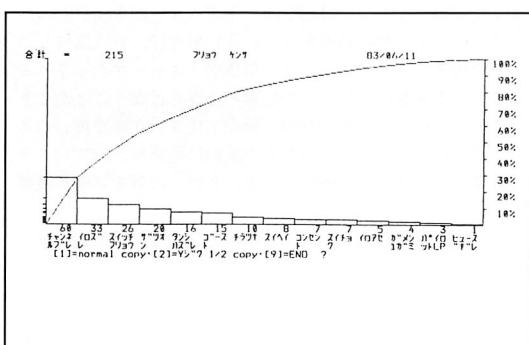
** UNIQUES-TQC (Ver 2) 総合メニュー ** 83/06/11

業務コードをどうぞ = 1

1. データ入力 [Xbar-R]
2. 度数データ入力 [Xbar-R]
3. データ入力 [P, U]
4. データ入力 [Pn, C]
5. 管理図作成
6. パレート図作成
7. 表題の登録及び保守
8. QCデータ 一覧表の作成
9. コピー 処理
99. 業務終了

データ入力 [Xbar-R] を 行います-OK=[RET], NG=[SP]

管理図メニュー

パレート図

開発会社名：㈱ユニ・システム
販　売　元：㈱ユニ・システム
問い合わせ先：㈱ユニ・システム
〠530 大阪市北区曽根崎新地1-3-25 毎日産業ビル４Ｆ
☎ (06)343-1086
価格：￥95,000
使用言語：BASIC＋マシン語
必要なOS：なし（F-BASIC V.4）
提供媒体：５インチ・ディスク　３枚
マニュアル：p.76
マニュアル別売：あり　￥20,000
サンプル・データ：なし（マニュアル）
出荷開始時期：'83年８月
最小システム構成：FM-11（２ドライブ・システム）
CRT
シリアル・ドッド・プリンタ

FM-11

SUPERPRO スーパープロ シリーズ 販売・仕入・在庫管理

SUPERPRO 販売・仕入・在庫管理は売上管理，売掛管理，仕入管理，買掛管理，商品管理，在庫管理が有機的に結合された，トータル・システムです。

本システムは処理件数においても，一定範囲内で自由に振り分けて使えるような，弾力性を持つ．

特徴

❶ 漢字で表示，印刷

簡単な音読み，訓読みの入力による，かな漢字変換を使い，得意先名，仕入先名，商品名，販売員名，住所などが漢字で表示，印刷できる．

❷ トータル・システム

売上，仕入，入金，支払，入出庫の伝票のデータを入力するだけで，売掛金管理，買掛金管理，粗利管理，および在庫管理ができる．

❸ 画面との対話型入力でやさしい操作

すべての処理は，画面に表示されているメニューから選択することで実行する．また，各処理においては，必要な手順はすべて画面に指示されるので，利用者は画面と対話しながら，簡単にデータ入力ができる．

❹ 売上伝票，請求書は（株）日本法令製を使用

売上伝票は販売MC-2．請求書は販売MC-6を使う．社名などの印刷も容易で，複写枚数も自由に設定できる．

また，仕入伝票，精算書についても同様．

❺ 締日に請求書を自動発行

締日には日付入力だけで，同一締日の得意先すべてに対して，請求書が自動発行される．

❻ 棚卸しも簡単に

棚卸しによる在庫数の補正が簡単に行なえる．

❼ 各種日報，管理資料も豊富

得意先，仕入先，商品，販売員ごとの各種日報，および管理資料も豊富に用意されている．これにより，1日の売上集計や報告書作成に時間がかかることもなくなる．

業務のサイクル

SUPERPRO 販売，仕入，在庫管理は，業務のサイクルによって分類すると，以下のようになる．

❶導入時業務

本システムを導入する際に必要な処理である．まず，本システムの使用条件（たとえば，小数使用の有無，丸め法など）を設定する．

その後，得意先，仕入先，販売員，商品の各台帳を作成する．

❷ 日次業務

本システムの中心となる部分で，毎日の売上伝票，仕入伝票の発行，入金，支払，入出庫の各データ入力，および各種日報の発行を行なう．

❸ 月次業務

本システムでは，各種データは，2つのサイクルによって管理されている．1つは締日から翌締日までの締日単位の管理，もう1つは月初めから月末までの月単位の管理である．

月次業務では，この月単位のデータを各種月報，管理表として発行する．

❹ 締日業務

請求書および精算書の発行を行なう．

❺ 随時業務

各種一覧表の発行を行なう．これら一覧表はいつでも，必要なときに出力することができる．

❻ 棚卸し業務

棚卸しを実施した際に，実棚数によって，計算機上の在庫数を補正するために行なう．

操作の実際

SUPERPRO 販売，仕入，在庫管理の操作が簡単であることを理解するために，ここでは売上伝票を説明する．

業務メニュー

画面に表示されているメニューの中から行ないたい処理のメニュー番号を入力するだけで実行できる．

まず処理のサイクルに対応した，主メニューが表示される．たとえば，売上伝票の処理を行ないたい場合は，日次業務なので，1を選択する．すんと，日次業務のメニューが表示される．

ここで，1を選択すれば売上伝票の処理を行なうことができる．このように『SUPERPRO 販売・仕入・在庫管理』の各プログラムは，メニュー画面からの選択に始まり，各プログラムの終了によりメニュー画面に戻る，といった形で処理が遷移する．

表1　販売・仕入・在庫管理で扱える件数

データの種類	最大件数
得意先台帳	あわせて1,000件
仕入先台帳	
商品台帳	
販売員台帳	100人
売上伝票枚数	あわせて 180枚/1日
仕入伝票枚数	

表2　各項目の桁数および範囲

項目の種類	範囲
得意先名	漢字で15文字以内
仕入先名	漢字で15文字以内
商品名	漢字で15文字以内
販売員名	漢字で10文字以内
単価	1,000万未満，小数点以下2桁
数量	1,000万未満，小数点以下3桁
金額	10億円未満

図1　売上伝票帳票

売上伝票

《掛売上》

様　昭和 58年 08月 23日

〒150
渋谷区宇田川町36－6
ワールド宇田川ビル
株式会社 リード・レックス
☎ 03（464）1241（代）

商品名	単位	数量	単価	金額
カラーテレビ　C－18K102	台	2	140,000	280,000
カラーテレビ　CV20T77F	台	2	200,000	400,000
ビデオ　レコーダー　F503	台	3	170,000	510,000
ビデオ　レコーダー　75BX	台	1	230,000	230,000
洗濯機　KW360M	台	1	80,000	80,000
摘要			合計	1,500,000

図2　売上伝票の画面ハードコピー

《売上伝票》　＊＊＊ 売上伝票 ＊＊＊　《掛売上》

得意先コード 1000
得意先名
郵便番号
住　所

日　付 58/08/23
伝票番号　1

	商品名	単位	数量	単価	金額
1	カラーテレビ　C－18K102	台	2	140,000	280,000
2	カラーテレビ　CV20T77F	台	2	200,000	400,000
3	ビデオ　レコーダー　F503	台	3	170,000	510,000
4	ビデオ　レコーダー　75BX	台	1	230,000	230,000
5	洗濯機　KW360M	台	1	80,000	80,000
	摘要			合計	1,500,000

確認して下さい。　OK〔PF1〕／変更〔PF5〕／キャンセル〔PF6〕

開発会社名：㈱リード・レックス
販売元：㈱リード・レックス
問い合せ先：㈱リード・レックス
〒150 東京都渋谷区宇田川町36-6　ワールド宇田川ビル
☎(03)464-1241
価格：￥200,000
使用言語：F-BASIC　V4.0
必要なOS：なし
提供媒体：5インチ・フロッピーディスク
マニュアル：あり，導入手引編，操作説明編，その他，計p.120
マニュアル別売：なし
サンプル・データ：なし
出荷開始時期：昭和58年9月
最小システム構成：FM-11EX(MB25050)またはFM-11AD(MB25040)またはFM-11ST(MB25030)いずれも，ミニフロッピー2ドライブ必要．
ディスプレイ：MB27311
プリンタ：ビジネス・プリンタ(MB27402)
プリンタ・ケーブル：MB26524
漢字ROMカード：MB28111

FM-11

DATA ACE

データ・エース

リレーショナル・データベース

DATA ACEは、パーソナル・コンピュータ・システムのため強力なリレーショナル・データベース管理システムである．社内の繁雑な業務をコンピュータ化させたいが，ソフトウェアを外注するにはコストが高く，なんとか自社内で開発したいと苦心している方や，簡易ソフトウェアに飽きたらず，データベースの機能をフルに活用して，データの整理，検索をパーソナル・コンピュータで実現させたいと考えている方々が，有効利用できるソフトウェア・パッケージである．

以下に特徴を示す．

❶項目の集合であるリレーション定義可能
❷リレーション間通信機能
❸データ統合機能
❹データ検索
❺カタログ機能
❻簡単な帳票作成，検索の自動処理
❼プログラム編集はスクリーン・エディット
❽データ操作はユーザーの自由なレイアウト，処理
❾グラフ表示可能

データベース利用のプロセス

初めに，データを蓄積するために，データベースを構築する．どういう項目のデータを管理するのかを，設定すればよい．次に，このデータベースにデータを入力し，まずは，検索業務や簡単な帳票をレイアウトして，データを活用することから始める．

さらに，蓄積されたデータを，より業務に適応したシステムとして，有効に利用するために，業務を分析しアプリケーション・システムを開発する．どのような帳票が欲しいのか，どのような内部計算が必要なのか，また，どのような入力画面が欲しいのかを設計し，具体化する．

このようなプロセスを経て，ユーザーは定型業務を完全にシステム化することが可能である．また非常型業務用にもデータを有効に利用することができる．

データベース定義機能(DDL)

データベース定義機能では、項目を定義し，それらを関係付けた，リレーションを定義してデータベースを構築します．また，実際のデータが入力された後でも，データ項目の長さ，フォーマット，リレーションに新たに項目を追加したり削除することができます．

リレーションの定義

項目の集合である，リレーションを定義する，たとえば，商品マスタを構築する場合には，ＨＩＮというリレーション名，商品番号を，キー項目，商品名などの項目をレイアウトし，各項目の入出力順を定義する．項目は，最大25個までレイアウトです．項目の合計の長さは255バイトまでである．また，ノーショナル機能と併用して，最高6段階までの多段階ソートが可能である．

リレーション間通信機能

離れた場所に置かれた，もう1台のDATA ACEシステムと，RS-232Cインターフェイスを介して通信することができる．

リレーション内のデータをレコード単位に送受信し，さらに受信側でリレーションの更新まで行なうという画期的な機能である．

音響カプラーを使えば，遠距離通信も可能となる．

データ検索

商品マスターの中から，商品番号を指定してデータを参照したり，任意の項目に条件を設定した複合条件検索が，簡単なコマンドで実行できる．このように，検索条件を設定できる項目が，自由に指定できるのが特長である．

カタログ機能(CAT)

カタログ機能は，プロシージャ，テキスト，プログラムの3つのモードを利用することにより，簡単な帳票の作成，文章登録，プログラムと編集が行なえる．

スクリーン・エディット機能を備えているので，効率よく画面上で編集できる．

プロシージャー
（簡単な帳票作成と検索の自動処理）

帳票タイトル，出力項目名，合計のような出力形式のレイアウトと，出力させるデータ項目などを設定できる．

データ検索を自動的に処理したい場合には，同様に検索条件を複数登録するだけで処理できる．

事務処理システムへの応用

コンピュータを利用して，一連の事務処理を行なうには，必ずプログラムが必要になる．例として，販売管理システムについて考えてみる．

1日の売上伝票を集計し，得意先台帳や商品台帳に記帳して，残高を更新する処理が必要である．1日の売上伝票の集計，あるいは各台帳ごとの集計処理はDATA ACEのDIL機能で，プ

図1　グラフ表示例

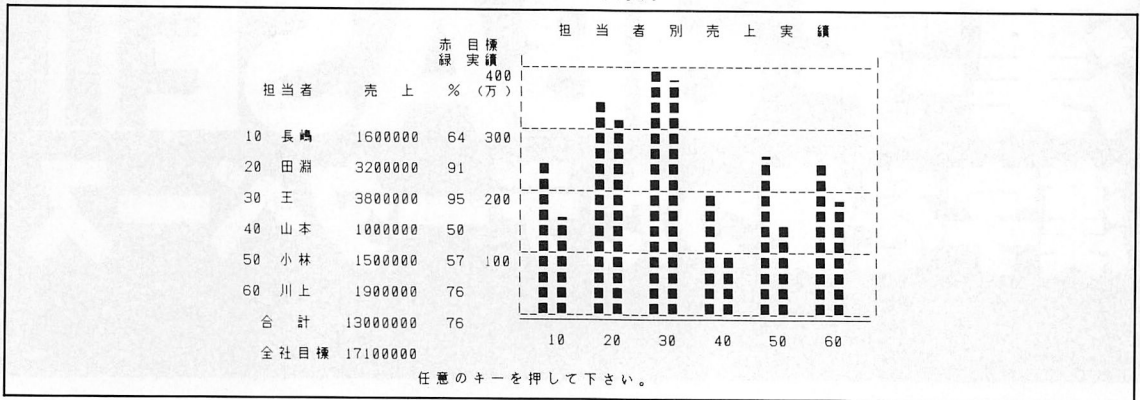



図2　ソート例

```
DIL> LIST HINS WHERE 仕入単価 < 50000 AND 今月売上数量 >= 10 ( HME STN KUS TZS )

商品名                  仕入単価  今月売上数量  当月在庫数
机上ライト               ¥4000           20          6

商品名                  仕入単価  今月売上数量  当月在庫数
電気コタツ              ¥15000           15          6

商品名                  仕入単価  今月売上数量  当月在庫数
ふとん乾燥機             ¥6000           10          2
```

図3　帳表出力例

*** 商品一覧表 ***　　日付 08/23/83　　PAGE : 1

品番	商品名	今月売上数量	今月売上金額	今月仕入数量	今月仕入金額	前月未在庫数	当月在庫数
1	14形カラーTV	10	¥890000	10	¥630000	3	13
2	机上ライト	20	¥100000	22	¥88000	4	6
3	電気コタツ	15	¥300000	20	¥300000	1	6
4	ふとん乾燥機	10	¥88000	8	¥48000	4	2
5	冷蔵庫	8	¥704000	10	¥620000	0	2
6	ラジカセ	32	¥1024000	50	¥1200000	5	23
7	電動タイプ	7	¥455000	10	¥500000	1	4
8	カーステレオ	38	¥4750000	30	¥2700000	8	0
10	FM-11	10	¥3980000	2	¥0	11	3
99	その他	6	¥124000	8	¥98800	2	4

FIELD-NAME	COUNT	SUM
今月売上金額	10	12415000
今月仕入金額	10	6184800

ログラムなしでできる．しかし，このように2つ以上の台帳に対しての複雑な処理は，プログラムで処理すべきことである．

まとめ

以上のように，DATA ACEは，データベース定義機能（DDL），データ統合機能（DIL），カタログ機能（CAT），データ操作機能（DML）の4つの強力な機能により，パソコン上で自由にデータを操作し，かつ自由なシステム開発を可能にする．

開発会社名：米国コンピューター・ソフトウェア・デザイン社
販売元：㈱ソフト工学研究所
問い合せ先：㈱ソフト工学研究所
〒101 東京都千代田区外神田3-11-2 ロックビル2F
☎03(251)1195
価格：￥275,000
使用言語：FORTH
OS：CP/M-86
提供媒体：5インチ・デスク
マニュアル：あり(約p.200)
マニュアル別売：なし
サンプル・データ：あり
出荷開始時期：'83年9月
最小システム構成：本体FM-11EX(MB25050)
CRT(MB27311またはMB27312)
プリンタ(MB27402)
漢字ROMカード(MB28111)
増設用ドライブ(MB27609)

FM-11

漢字dBASEⅡ 漢字を使ったデータベース

dBASEⅡは、従来大型コンピュータで使われてきた、データベースをパーソナルコンピュータ・レベルで利用できるようにしたものである．専門的知識を必要とせず，初心者でも8個の基本コマンドでデータの登録，修正，検索が簡単に行なえる．

そして，複雑なデータ処理を行なう命令や，スクリーン（プリンタ）に対する命令が強力であり，いままでのBASICのものと比べて約1/10以下の工程でプログラムが作成できる．

データは，CP/Mなどの共通OS上で動く，他のソフトウェアのデータと互換性がとれる機能が備わっており，BASICでデータを処理し，dBASEⅡでこれを管理するということもできる．

データの互換性は，ソフトウェア間だけでなく，CP/Mのデータを CP/M-86で処理することもできる．

また，文字ストリング演算や，マシン語ルーチンの使用などの機能もある．

dBASEⅡの言語体系

予約語は，入力した文字だけがチェックされ，エラーが発見されると，エラーメッセージを表示して，エラーの場所を示すとともに訂正するか否かを聞いてくる．

必要に応じてエラー箇所を直し，長い命令を何度も入力することがないようになっている．

さらに，マクロ機能(&)により，メモリ変数に命令，あるいは命令の一部を定義して実行できる．

また，各命令のオプションは，決った順序などなく，任意の順に指定できる．

```
例
COPY TO TEMP FOR 単価>=1000 FIELD コード，
単価，数量
➡ COPY FOR 単価>=1000 FIELD コード，単価，
数量 TO TEMP
```

この機能により，ユーザーは命令のシンタックスなど気にせずに使用できる．

データベースに必要不可欠なデータ検索は FOR〈条件〉で行なうことができる．この条件が真(True)となるものだけが命令の対象となる．

```
例
REPORT FORM ～ FOR 住所="渋谷区"
フィールド"住所"が渋谷区のデータだけレポートする．
```

条件は，AND，OR，NOTなどを使って，より複雑な条件とすることもできる．さらに，文字列検索($)や，関数も条件の中に使える．

構造化プログラムのための命令はループ(DO WHILE～END DO，条件判断(IF～ELSE～ENDIF，DO CASE～ CASE～……OTHER WISE～ENDCASE などがある．

ファイルの更新

トランザクション・ファイル(BFILE)にデータ(数量)を入力し，そのデータ(数量)でマスタ・ファイル(AFILE)を更新することができる．

使用ファイルの構造は表2，図1の通りである．

❶トランザクション・ファイルのソート

BFILEのコードでソートを行なう．

```
USE BFILE SORT ON コード TO SF
```

ソートされたファイルをSFとする．

❷同じコードで集計

SFファイルの同じコードを集計する．

```
USE SF TOTAL ON コード TO S
```

集計されたファイルをSとする．

❸ マスタ・ファイル更新

コードで集計されたSファイルでマスタ・ファイルを更新する．

```
USE AFILE UPDATE FROM S ON
コード ADD 数量
```

ファイルの結合

2つのファイルの項目を結合し，新しいファイルを作成することができる．

使用ファイルの構造は以下の通りである．

```
SELECT PRIMARY USE AFILE SELECT SECONDA
RY USE BFILE SELECT PRIMARY JOIN TO CFILE
FOR   コード=S．コード FIELD
      コード，商品名，単価，数量
```

作成されるファイル(CFILE)の構造は，コード，商品名，

表1　dBACEII　基本コマンド

CREATE	ファイルを作成
USE	使用宣言
APPEND	データ追加
EDIT	データ修正
INDEX	インデックス
COPY	コピー機能
RFPORT	レポート機能
QUIT	終了

表2　使用ファイルの構造例1

AFILE

コード	C	4
数　量	N	10
⋮		

BFILE

コード	C	4
数　量	N	10

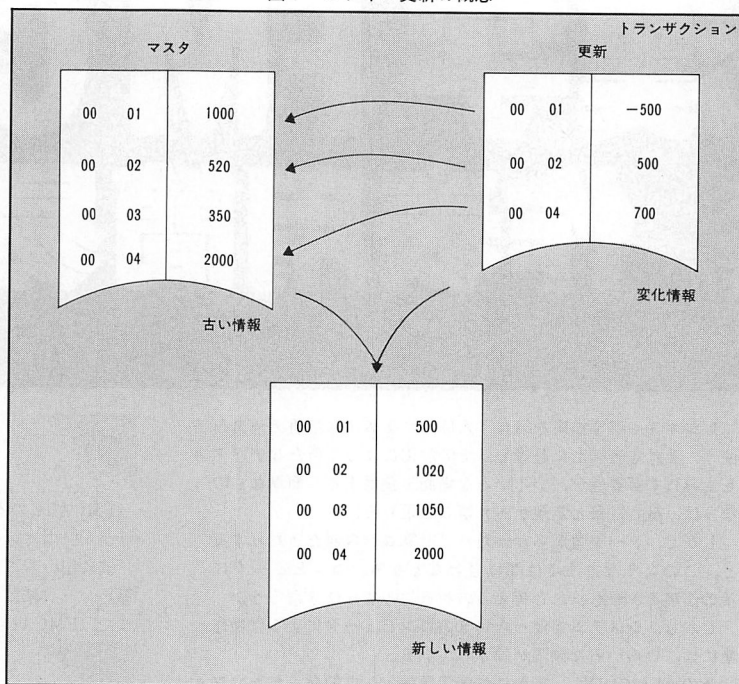
表3　使用ファイル構造例2

AFILE

コード	C	4
商品名	C	30
単　価	N	7

BFILE

コード	C	4
数　量	N	5

図1　ファイル更新の概念

単価，数量となる．

2つのファイルの中に同じフィールド名があるときは，〝S.〟(あるいは〝P.〟)をフィールド名の前に付ける．〝S.コード〟はSECONDARYエリアのフィールドとみなされる．

ファイルの構造の変更

データベース・ファイルの構造を変更するとき，データはすべてクリアされるので，別のファイルにバックアップをとっておく．

ただし，フィールド名は変更しないものとする．

```
USE AFILE
COPY STRUCTURE TO
    TENP
USE TENP
MODIFY STRUCTURE
        ⸗
     構造　修正
        ⸗
APPEND FROM AFILE
COPY TO AFILE
USE AFILE
DELETE FILE TEMP
```

構造を修正するファイルをAFILEとし，ファイル構造のみをTEMPファイルにコピーする．そのワーク・ファイル上で，構造修正を行ない，AFILEのデータを追加する．

次に，ワーク・ファイルをAFILEにコピーし作業を終了する．

dBASEは，フィールド名によって処理されるので，テキスト・ファイル(TEXT・TXT)にデータをコピーして，構造を修正した後に追加する．

データ追加，修正

ユーザーが作成したスクリーン上で，入力を行ない，データをファイルに追加，修正する．

```
        ⸗
  データ入力（修正）
        ⸗
  〔APPEND BLANK〕
   REPLACE……
        ⸗
```

データ入力で，画面制御命令(ERASE，@など)でスクリーンを描き，入出力を行なう．

修正する場合は，APPEND，BLANK命令を省略し，ポインタの移動を行なう．ポインタの移動は，LOCATE，FIND，GOTO，SKIP，CONTINU命令などで行なう．

スクリーン上では，メモリ変数に入力し，REPLACE命令でデータを置き換える．

開発会社名：Ashton-Tate
総販売元：ソフトウェア・インターナショナル㈱
問い合わせ先：ソフトウェア・インターナショナル㈱インフォメーション・センター
〒東京都渋谷区神宮前6-12-20東洋ビル9Ｆ
☎(03)486-7155
価格：￥268,000(漢字バージョン)
使用言語：アセンブラ
必要なＯＳ：ＣＰ/Ｍ-86
提供媒体：５インチ・ディスク
マニュアル：あり，p.160
マニュアル別売：なし
サンプル・データ：あり
出荷開始時期：'83年６月１日
最小システム構成：FM-11EX　RAM 128K
フロッピーディスク１台

FM-11

漢字FM CALC エフエム・カルク
作表用簡易言語

ビジネスや科学技術を行なう人は，さまざまな数値データを使う．それを表の形に整理し，その加工によって新たなデータを生みだす必要性が，いろいろな場面で登場する．簡単なものならば，紙と鉛筆と定規があればこと足りる．

しかし，データ量が多かったり，計算式が複雑だったりすると，このような方法では間に合わなくなる，コンピュータにその仕事をさせたい，と考えるのが自然の成りゆきだろう．

しかし，システムを使った場合のコンピュータによる作表作業には，いろいろな制限が課されている．

その点FMCLCは，純粋に作表用言語として開発されたソフトウェアなので，

❶BACISやFLEX，CP/MなどのOSを必要としない．
❷POWER ONで起動する．
❸起動すると作表画面が，いきなりディスプレイに表示される．

という特徴を持っている．

表の構造

起動時に表示された画面は，FMCALCの表の一部であって，列名A～Zの26列(コマンドにより，A～Z, a～zの52列まで拡張できる)，行数128の巨大な表の大部分は現在隠れているにすぎない．画画の移動により，希望の領域を表示させることができる．

列名(A～Z)と行番号(1～128)によって決まる領域をブロックと呼び，FMCALCの入力単位となる．その列名と行番号が座標にあたり，たとえばA1，C23，Y105などと表し，ブロックの位置を決定する．

現在の入力対象ブロックは，ブロック・カーソルにより示される(起動画面においてA1位置にある■■■がブロック・カーソル)．データなどを入力し⏎キーを押すと，ブロック・カーソルのあるブロックにデータが入り，表示される．ブロック・カーソルの幅は，初期時には14に設定されているが，これは，1～59の間で変更ができる．ブロック・カーソルの移動は，カーソル移動キーが利用できるほか，ブロック座標の指定による方法もある．また，CURSORコマンドも，効率的なブロック・カーソルの移動に有用である．このブロックには文字（漢字を含む），数値式のいずれかを入力できる．

文字の入力

文字は半角文字と全角文字がある．全角文字には，キーボードから直接入力する英字，カナ，ひらがな，記号と，カナ・漢字変換による漢字入力とがある．

式の入力

FMCALCの有力な武器は，四則演算や各種関数を用いた計算が，自由に行なえることである．

式に用いることのできる演算子は以下のものである．＋（加算），－（減算），＊（乗算），／（除算），＾（べき乗），また，FMCALCで利用できる関数は，表2の通りである．

SORT

これ以外のコマンドは表をみれば，簡単にわかる．作表に有効なSORTに関して，例をあげて設明する，SORTは，指定した行，列に基づいて(KEY行，KEY列と呼ぶ)，行，列を並べかえるものである．

図2は，商品1～17のそれぞれについて，　地域別の売上と合計を記録したものである．

SORTコマンドにより，合計列(J列)を KEY として行を，数値の大きい順に並べかえると図3のようになる（範囲はA1，J17)

TOTAL 行を除いて，行の並べかえが行われ，合計列の大きい順になっていることがわかる．本列では大きい順だが，小さい順にすることもできるし，文字の場合のアルファベット，五十音順の並べかえも可能である．

実行に必要なパラメータは，コマンドにあらかじめ指定するのではなく，⏎を押すことで，システムが表示するメッセージに応答する形式をとっている．

このコマンドは，FMCALCの中でもひときわ有力な機能で，データ整理，分折に威力を発揮する．

数値列（行）を KEY にする場合は，日付順の並べかえもできる．またSORTの結果を利用して，メジアンを計算することも容易である．文字データ列（行）を KEY にした場合，名簿作成などに効果がある．ランダムに入力したものを五十音順にSORTすれば，そのまま名簿が作成できる．

表1　漢字FMCALCの仕様

項目	内容
プログラミング言語	マシン語.
プログラム・サイズ	約30Kバイト.
データ・サイズ	約60Kバイト.
漢字辞書サイズ	約20Kバイト
供給媒体	5インチフロッピーディスク
コマンド	19種
PFキー登録	10種(オプション付きを含む).
表の大きさ	横52×縦128(最大).
項目幅	1～59文字(スタート時は14文字)
使用文字(半角)	カタカナ, 英文字, 数字, グラフィック記号.
(全角)	漢字, ギリシャ文字, 各種記号, ひらがな, カタカナ
入力方式	ローマ字.
漢字入力方式	カーソル移動方式.
入力文字	漢字一文字ごとのカナ・漢字変換
入力数値	左づめ表字, 最大59文字/項(全角) 29文字
算式	右づめ3桁ごとにカンマ付き表示.
組み込み関数	数字式通り. ( )付き. 最大60文字 一般関数　10種. 特殊関数　2種
数値精度	一般式　16桁. 関数　13桁. PAI定数16桁.
計算モード	非計算モードを含め3種.
プリント・アウト	並通文字.
データ・ファイル	ドライブ0またはドライブ1(立ち上げ時に指定), BASICのDATAと同形式.

表2　FMCALCの関数

	関数名	目的	書式	精度	備考
	SIN	正弦を与える.	SIN(<引数>)	13	<引数>の単位はラジアン.
	COS	余弦を与える.	COS(<引数>)	13	<引数>の単位はラジアン.
	TAN	正弦を与える.	TAN(<引数>)	13	<引数>の単位はラジアン.
一般関数	ATN	逆正接を与える	ATN(<引数>)	13	<引数>の単位はラジアン. 数値範囲は$-\pi/2$から$\pi/2$まで.
	ABS	絶対値を与える.	ABS(<引数>)	16	
	EXP	eのべき乗を与える.	EXP(<引数>)	15	
	LOG	自然対数を与える.	LOG(<引数>)	15	<引数>は正.
	SRQ	平方根を与える.	SQR(<引数>)	—	<引数>は0または正.
	INT	整数化を行なう.	INT(<引数>)	—	<引数>をこえない最大の整数を与える.
	SGN	符号を与える.	SGN(<引数>)	—	<引数>が正…………1. <引数>が0…………0. <引数>が負…………−1.
特殊関数	SUM	加算を行なう.	SUM(<座標1>, <座標2>)	16	座標1から座標2までの加算.
	AVR	平均を求める	AVR(<座標1>, <座標2>)	16	座標1から座標2までの平均.

表3　コマンド一覧表

コマンド名	内容
BLNK	項目の内容を消去する.
INIT	全項目のデータを消去する.
COPY	式, フォーマット文字を含んだデータをコピーする.
CURSOR	データ入力後のカーソルの移動方向を設定する.
DELETE	行または列を削除する.
FX	式をプリンタにプリントする.
FILES	ドライブ上のファイル名を表示する. またファイルの消去もする.
LOCK	タイトル項目の行, 列を固定し, スクロール中に消えないようにする.
FORMAT	数値データの書式を設定する.
KANJI	漢字を表示をON/OFFする. OFFの場合スクロールなどが速くなる.
MEMORY	メモリの残量を表示する.
INSERT	行または列を挿入する.
LOAD	ディスク上のファイルをロードする.
PRINT	集計上のデータをプリンタに出力する.
SAVE	集計上のデータをディスクに格納する.
SORT	指定された行, または列のデータを並べ変える.
TITLE	表のタイトル名を表示する.
WIDTH	各項目の幅を設定する. またプリンタの種類も設定する.
#	計算を実行する. また計算モードを設定する.

図1

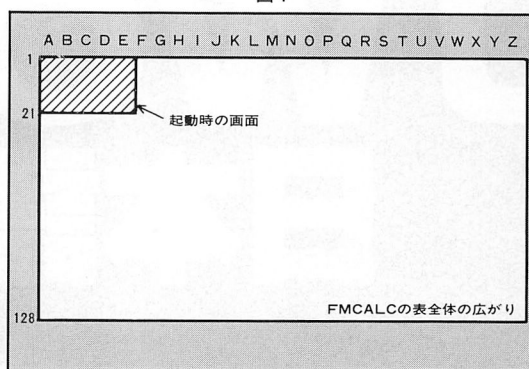



図2

	秋葉原	新宿	箱館	大阪	広島	博多	合計
商品 1	1,184	1,119	1,189	2,278	2,738	2,183	10,691
商品 2	1,274	2,738	1,678	2,739	1,738	3,820	13,987
商品 3	1,637	2,189	1,897	2,738	1,836	2,673	13,015
商品 4	1,839	1,738	1,354	2,738	1,725	2,740	12,134
商品 5	1,803	1,980	1,245	1,830	1,783	1,947	10,588
商品 6	1,279	1,728	1,787	1,830	1,728	1,836	10,188
商品 7	1,280	1,738	1,870	2,739	1,942	3,172	12,741
商品 8	1,893	1,849	1,678	1,479	1,528	2,462	10,889
商品 9	1,782	3,783	1,426	1,179	1,782	2,562	12,514
商品 10	1,042	2,849	1,674	3,729	2,643	1,947	13,884
商品 11	1,801	2,378	1,637	2,362	2,739	3,184	14,101
商品 12	2,673	2,748	1,782	2,638	2,739	1,856	14,436
商品 13	2,892	2,539	1,935	2,489	2,135	1,987	13,977
商品 14	2,183	2,991	1,637	1,352	2,365	2,645	13,173
商品 15	2,463	3,820	2,163	1,639	1,991	3,251	15,327
商品 16	2,749	3,126	1,468	3,263	1,247	2,553	14,406
商品 17	2,138	3,729	1,673	1,739	2,153	3,364	14,796
合計計	31,912	43,042	28,093	38,806	34,812	44,182	220,847

A B C D E F G H I J

図3

	秋葉原	新宿	箱館	大阪	広島	博多	合計
商品 15	2,463	3,820	2,163	1,639	1,991	3,251	15,327
商品 17	2,138	3,729	1,673	1,739	2,153	3,364	14,796
商品 12	2,673	2,748	1,782	2,638	2,739	1,856	14,436
商品 16	2,749	3,126	1,468	3,263	1,247	2,553	14,406
商品 11	1,801	2,378	1,637	2,362	2,739	3,184	14,101
商品 2	1,274	2,738	1,678	2,739	1,738	3,820	13,987
商品 13	2,892	2,539	1,935	2,489	2,135	1,987	13,977
商品 10	1,042	2,849	1,674	3,728	2,643	1,947	13,884
商品 14	2,183	2,991	1,637	1,352	2,365	2,645	13,173
商品 3	1,637	2,189	1,897	2,783	1,836	2,673	13,015
商品 7	1,280	1,738	1,870	2,739	1,942	3,172	12,741
商品 9	1,782	3,783	1,426	1,179	1,782	2,562	12,514
商品 4	1,839	1,783	1,354	2,738	1,725	2,740	12,134
商品 8	1,893	1,849	1,678	1,479	1,528	2,462	10,889
商品 1	1,184	1,119	1,189	2,278	2,738	2,183	10,691
商品 5	1,803	1,980	1,245	1,830	1,783	1,947	10,588
商品 6	1,279	1,728	1,787	1,830	1,728	1,836	10,188
合計	31,912	43,042	28,093	38,806	34,812	44,182	220,747

A B C D E F G H I J

開発会社名：ソフトマート㈱
販　売　元：ソフトマート㈱
問い合せ先：ソフトマート㈱
〒101 東京都千代田区外神田須田町1-18-6第1谷ビル1F
☎(03)256-5881
価格：¥58,000
使用言語：アセンブラ
必要なOS：なし
提供媒体：5インチ・ディスク
マニュアル：あり, p.200
マニュアル別売：あり
サンプル・データ：なし
出荷開始時期：'83年　9月
最小システム構成：FM-11AD(orEX)
400ラインCRT
漢字ROMカード
プリンタ(コピー時)

FM-11
JWORD（ジェイワード）日本語ワープロ

『JWORD』はパーソナル・コンピュータ上で動く，日本語ワードプロセッサ・ソフトウェアである．JWORDは，抜群の操作性と高速性を持ち，専用ワープロ機なみの辞書を備えた，使いやすいソフトウェアで，JWORDを使うと，FM-11が日本語ワープロに変身するのである．

以下に特徴をまとめてみる．

❶ 文節単位によるカナ漢字変換
❷ 65,000語の単語辞書内蔵
❸ ローマ字，混在など4入力方式
❹ 強力な作表，演算機能装備
❺ アセンブラ記述で超高速動作
❻ 2段階メニュー方式で高速操作可
❼ CP/M標準ファイル使用で発展性有

機能

JWORDを起動すると，初期メニューが表示される．JWORDで実行できることは，この初期メニューの内容である．

かな-漢字変換方式

JWORDは，かな漢字変換方式を採用している．変換される文字列（読み）は，次の形式で入力される．

❶ カタカナ入力
❷ ローマ字入力
❸ 混在入力（カタカナとローマ字の混在）
❹ 一文字入力（F9，F10キーを使用）

変換キーは次のキーを使って，漢字，カタカナ，ひらがななどに変換できる．

1）．F1 ……単語に変換
2）．F2 ……ひらがなに変換
3）．F3 ……カタカナに変換
4）．F4 ……漢字1字に変換
5）．F5 ……入力のまま
6）．F6 ……同一キーの逆戻り
7）．F7 ……半角文字に変換

JWORDは，文節変換を行なうので，単語（複合動詞および活用を含む）とそれに続く『ひらがな』の語をF1キーの単語変換で，1度に変換できる（モウシアゲマス→申し上げます；モウシアゲマシタ→申し上げました；モウシアゲタトコロ→申し上げたところ）．

文書の作成，更新

初期メニューで，『作成』を選択し，ファイル名およびメモを入れ，⏎キーを押すと図2の画面になる．この画面では，メニューに表示されているコマンドが入力可能で，新規に作成するときはIキーを押すことで文書が作成できる．

すでに作成してある文書を更新する場合は，更新を使用する．更新はファイル名の一覧が画面に表われるので，該当ファイルにカーソル移動し選択する．

コマンド

文書の編集に使用するコマンドは図3の通りであるが，コマンド・キーを押した後の，いかなる状態でもESCキーを押すことにより，コマンド入力状態に戻る．また，コマンド・キーは，キーボード上のアルファベット・キーを意味し，キーを押すとさらに詳細なメニューが表示される．

文書の印刷

初期メニューで印刷を選択すると図4の画面が表示される．詳細な指定を選択すると図5の画面が表われ，自由に変更ができる．

文書ファイルの扱い

初期メニューでファイル名（一覧，削除，名前変更）を選択するとファイルが表示される．上段のメニューに従って操作すると，ファイルを削除したり，名前を変更したりできる．

作表・演算

初期メニューで『作表・演算』を選択すると，カーソルの移動のみで作表ができ，電卓形式の演算もできる．また，行，列の挿入，削除も簡単に行なえる．

その他

その他の機能としては，単語辞書（8インチ版では65,000語＋ユーザ登録語，5インチ版では24,000語＋ユーザ登録語）のすべての単語が表示，印刷ができ，不要な単語を削除することもできる．また，宛名だけを変更して，印刷するといったとき便利な，差し込み印刷ができる．

FM-11ではユーザが独自にパターンを作成するユーティリティ（GAIJI）が用意されている．このユーティリティにより作成されたパターンに読みをつけ，文書作成中にその読みと単語変換を用いれば，表示，印刷ができる機能もある．

図1　初期メニュー

```
                                        1983年 6月27日

■■■■■ JWORD ■■■■■■■■■■■■■■■■■■■■■■

        ◆  作 成
        ◇  更 新
        ◇  印 刷
        ◇  終 了
        ◇  フ ァ イ ル 名 （一覧・削除・名前変更）
        ◇  作 表 ・ 演 算
        ◇  単 語 登 録
        ◇  単 語 削 除
        ◇  外 字 登 録
        ◇  単 語 辞 書 表 示

■■■■■■■■■■■■■■■■■■■■■■■ ACEL ■■■■■

  ↑↓キーでカーソルを動かし、<CR>を押して下さい。
```

図2

```
+コマンド: I挿入 X訂正 D削除 S置換 F検索 P送り J跳ぶ A位置
  H再表示 M移動 C複写 B除去 R読込 W書出 K制御 L字形 T単語 E終了
1 . . . : . . . . 10. . . . : . . . . 20. . . . : . . . . 30. . . . : . . . . 40
```

図3　コマンド一覧

コマンド	意味	内容
A	位置設定(At)	範囲を設定します.
B	除去(Block delete)	文字列を除去します.
C	複写(Copy)	文字列を複写します.
D	削除(Delete)	文字列を削除します.
E	編集終了(Exit)	編集を終了します.
F	検索(Find)	文字列を検索します.
H	ヘルプ(Help)	メニューを表示します.
I	挿入(Insert)	文字列を挿入します.
J	跳ぶ(Jump)	指定位置へ跳びます.
K	制御キー(Key)	センタリング，右づめなどを行ないます.
L	字形(List control)	倍角文字にします.
M	移動(Move)	文字列を移動します.
P	画面送り(Page)	画面の前・後送りをします.
R	読み込み(Read)	他のファイル読み込みます.
S	置換(Subsitute)	文字列を置換します.
T	単語登録(Tango)	単語を登録します.
W	書き出し(Write)	文字列を他のファイルへ書き出します.
X	訂正(eXchange)	文字列を訂正します.

図4　印刷モード

ファイル名　：　XXXXXX

用紙のサイズ：　3　（0：A4　1：B5　2：B4）

（3：10×11吋　4：15×11吋）

詳細な指定　：　0　（0：しない　1：する）

図5　詳細を指定する

◆　詳細な印刷情報の指定（A4の場合の例）

印刷部数	:	1	（1－999）
印刷開始ページ	:	1	（1－999）
印刷終了ページ	:	999	（1－999　999：最後まで）
ページ番号開始値	:	0	（0－999　0：ページ番号なし）
用紙の種類	:	0	（0：連続　1：単票）
1ページの行数	:	40	（20－80行）
1行の文字数	:	3	（0：24字　1：34字） （2：40字　3：50字） （英字：文字間ピッチ（A＝0…Z＝25））
上部余白	:	0	（0－78行）
左部余白	:	0	（0－48行）
横書き・縦書き	:	0	（0：横　1：縦）
差込みファイル名	:		

図6　印字例(1) 61/100

転勤の挨拶状

拝啓　新緑さわやかな季節となりました。

ご健勝にお過しのこととお喜び申し上げます。

さて、此の度札幌支店営業部勤務を命ぜられこのほど無事着任いたしました。

本社計画部勤務中は、公私にわたり、ひとかたならぬご指導ご厚情を賜り、誠に有難うございました。心から御礼申し上げます。

今後ともより一層のご支援を賜りますよう、お願い申し上げます。

敬具

昭和五十八年四月

勤務先　☆☆☆☆自動車工業株式会社　札幌支店

〒060　札幌市中央区北一条西一丁目

丸量　一郎

自宅　〒062　札幌市豊平区月寒東一条

図7　印字例(2) 55/100

転勤の挨拶状

拝啓　新緑さわやかな季節となりました。

ご健勝にお過しのこととお喜び申し上げます。

さて、此の度札幌支店営業部勤務を命ぜられこのほど無事着任いたしました。

本社計画部勤務中は、公私にわたり、ひとかたならぬご指導ご厚情を賜り、誠に有難うございました。心から御礼申し上げます。

今後ともより一層のご支援を賜りますよう、お願い申し上げます。

敬具

昭和五十八年四月

勤務先　☆☆☆☆自動車工業株式会社　札幌支店

〒060　札幌市中央区北一条西一丁目

丸量　一郎

自宅　〒062　札幌市豊平区月寒東一条

開発会社名：エイセル㈱

販売元：エイセル㈱

問い合わせ先：エイセル㈱

〒102 東京都千代田区麹町3-4-7 ビルディング啓

☎(03)263-6421

価格：￥82,000(5インチ，8インチとも)

使用言語：アセンブラ

必要なOS：CP/M-86

提供媒体：5インチ・ディスク

8インチ・ディスク

マニュアル：あり，￥8,000/冊

サンプル・データ：なし

出荷開始時期：昭和58年8月末日(バージョン2.0a)

最小システム構成：本体FM-11EX

フロッピィー5インチ×2または8インチ×2

プリンタ：ビジネス・プリンタ

その他：バックアップ・フロッピー1部無償添付

フロッピィー破損交換：￥5,000

バージョンアップ交換：￥10,000

FM-11
COMAS-WPE1 日本語ワードプロセッサ

COMAS-WPE 1 は，FM-11を対象として開発した，5 インチ版日本語ワードプロセッサである．FM-7，8 用で好評のWPV2の機能をすべて受け継ぎ，さらにFM-11の大容量，高速処理能力を活用して，機能，操作性の向上を図っている．

その特徴としては，以下のような点があげられる．

❶CP/MなどのOSを必要とせず，68B09Eの制御下でのオート・スタートにより使用できる．
❷当社独自の簡易文節変換方式の採用により，文書の入力，変換が非常に容易かつ迅速にできる．
❸FM-11の大容量，高速処理能力を活用することにより，文書切り貼り，文字列検索ができるなど，編集機能も向上し，キメ細かくスピーディーな文書作成ができる．
❹当社のワープロ・ソフトとしては，上位の機能をもちながら，価格が¥55,000と手ごろである．

機能概要

入力方式

漢字・熟語変換のための入力方式としては，次の 4 種がある．

❶カナ入力
❷ローマ字入力（ヘボン式および訓令式）
❸ローマ字カナ混在入力
❹JISコード入力

以上 1 ～ 3 の入力方式に対しては，簡易文節変換による漢字熟語への変換を行なう．

変換方式

カナ・ローマ字の入力に対する漢字・熟語の変換は，簡易文節変換による．これは，1 つの漢字，または熟語から次の漢字，または熟語の前までの読み入力について，1 度に漢字かな混じりの文節に変換する機能である．

たとえば，次のような例文について，従来の熟語単位の変換と比較（当社比）してみると，その効率のよさが，おわかりいただけると思う．

例文：文書を作りたいときの手順について

❶従来の変換方式での入力

次のように送りがなの訂正を含め，7 段階の入力，変換操作が必要となる．

ぶんしょ／を／つくる／りたい／ ときの／てじゅん／について

❷簡易文節変換での入力

次のように送りがなの訂正もなく，3 段階の入力，変換操作で済む．

ぶんしょを／つくりたいときの／てじゅんについて

この簡易文節変換は，ローマ字，またはかなローマ字混在の入力に対しても機能する．また，同音異義語が変換された場合の再変換は，漢字・熟語のみが対象となる．

さらにWPE1では，最終使用優先学習方式による漢字変換のため，漢字・熟語の読みに対する変換は，前回のものが最初に変換，表示される．

その他の変換機能

上記の簡易文節変換のほか，次のような変換機能がある．

1）バック変換
2）無変換
3）カタカナ変換

この他，ローマ字入力に対する，ひらがな変換，カタカナ変換機能がある．

使用字種

❶JIS第 1 水準のすべての漢字，非漢字が使用できる．
❷熟語ファイルには，漢字，記号を含め27,000語以上の熟語が登録されている．
❸別売のサポート・ソフトを用いると，熟語ファイルの熟語追加，削除，外字の登録，削除ができる．
❹JIS第 1 水準外の文字，記号を持ったプリンタを使う場合には，JISコード入力によって，その文字，記号を印刷させることができる．

文書処理能力

1 ページ40字×70行までの設定で，1 文書は最大320行まで作成できる．

1 枚の文書ファイルには，1 文書320行として 8 文書まで（A 4でp. 50～60）を登録することができる．

画面表示

画面に表示される文書は，CRTにより，

640×400ドット：40字×18行
640×200ドット：40字×10行

が表示される．

画面のスクロールは，1 行または10行単位での上下スクロールを行なうことができる．

図1

第104期 決算公告

昭和57年06月30日

新宿区新宿3丁目18番5号
○ ○ 株式会社
(57年 3月31日現在)

貸借対照表

資産の部		負債 および 資本の部	
資産の部	2,688,746,450	負債の部	1,688,954,350
流動資産	1,572,993,960	流動負債	1,127,401,920
現金預金	116,454,270	支払手形	246,554,390
受取手形	247,400,470	買掛金	337,004,280
売掛金	467,357,170	短期借入金	104,386,600
有価証券	207,243,860	未払金	247,169,630
商品・製品・半製品	273,146,470	未払費用	109,714,240
仕掛品	36,432,550	事業税等引当金	19,177,120
原材料・貯蔵品	83,279,370	法人税等引当金	56,686,400
前払費用	2,447,580	その他の流動負債	6,709,260
未収入金	100,504,070	固定負債	554,752,430
その他の流動資産	48,908,110	社債	68,432,500
貸倒引当金	△ 10,240,000	長期借入金	32,828,420
固定資産	1,115,752,490	預り保証金	78,712,350
有形固定資産	498,430,070	退職給与引当金	358,445,310
建物・構築物	331,667,370	その他の固定負債	16,333,850
機械装置	752,717,790	特定引当金	6,800,000
車両・器具備品	83,537,840	価格変動準備金	6,800,000
減価償却引当金	△ 809,588,500	資本の部	999,792,100
土地	75,427,120		
建設仮勘定	70,028,440		
無形固定資産	7,424,630	資本金	166,175,710
工業所有権	492,260	法定準備金	319,176,440
施設利用権	6,932,370	資本準備金	280,327,490
投資等	609,897,790	利益準備金	38,848,950
投資有価証券	209,335,780	剰余金	514,439,950
子会社株式	94,417,780	株主配当引当積立金	69,700,000
子会社・出資金	70,163,260	従業員退職手当積金	14,600,000
長期貸付金	218,794,470	特別償却準備金	22,640,000
長期前払費用	4,051,940	海外投資損失準備金	9,610,000
その他の投資等	10,454,560	別途積立金	287,200,000
貸倒引当金	△ 2,680,000	当期未処分利益	110,689,950
合計	2,688,746,450	合計	2,688,746,450

画面編集機能

紙面の都合で詳述できないが，主なものを列挙すれば次のようになる．

❶1ページの行数設定，変更
❷文書訂正（上書き）
❸文字列挿入，削除
❹行の挿入，削除
❺文字列複写
❻短文登録，複写，削除
❼文書挿入（文書切り貼り）
❽左右マージンの設定，変更
❾タブ位置の設定，変更
❿文字列のセンタリング
⓫文字列の右詰め，左詰め
⓬文字列検索
⓭縦，横罫線，枠組み
⓮横倍角文字指定
⓯半角文字指定（英，数字，記号）
⓰アンダーライン指定

印刷機能

印刷機能についても，列挙すれば次の機能がある．画面編集機能と合わせて，きめ細かな文書作成ができる．

❶用紙サイズ指定
❷文字間隔指定
❸行間隔指定
❹単表印刷(ビジネス・プリンタのみ)
❺袋とじ印刷(ビジネス・プリンタのみ)
❻縦書き，横書き指定
❼印刷部数指定
❽印刷開始ページ指定
❾ページ印刷指定

図2

御見積書

見積書No.

昭和　年　月　日

殿

○ △ 商事株式会社
東京都渋谷区渋谷3−18−5
TEL (407) 8893
代表取締役

先にご照会いただきました

の件につき，下記の通りお見積り申し上げます。

1. 御見積金額計
2. 納期
3. 納入場所
4. 見積有効期限
5. 御支払条件等

品名内訳	数値	単価	金額

図3

開発会社名：㈱コマス
販売元：㈱コマス　パソコン事業部
問い合せ先：㈱コマス　パソコン事業部
〒150 渋谷区渋谷1-14-6 平野屋ビル
☎(03)407-8893
価格：¥55,000
使用言語：アセンブラ
必要なOS：なし
提供媒体：5インチ・フロッピーディスク
　WPE1システム・ディスク　1枚
　熟語ファイル・ディスク　1枚
　操作説明書(マニュアル)　1枚
マニュアル：あり　約p.80，別売せず
サンプル・データ：なし
最小システム構成：FM-11（薄型ミニフロッピーディスク増設のもの）
　漢字ROMカード
　CRTディスプレイ(640×400ドットまたは640×200ドット)
　プリンタ（次のものが使用できる）
　富士通MB27401〜27406, 27410およびビジネス・プリンタ，エプソンMP-80TYPE2, 3, FP-80, RP-80, MP-80K, MP-130K, UP-130K，セイコウGP-550E，スター精密DPX-510, 510J，関東電子KP-3000
出荷開始時期：'83年9月19日

FM-11

ザ・グラフィックツール

GT-11 THE GRAPHIC TOOL

コンピュータ・グラフィックスに挑戦

GT-11は多くの人がパソコンのグラフィック機能を，フルに活用して，絵，図，文字を簡単に描けるように開発した，グラフィック・ツール・ソフトである．

オペレーションを覚えるだけで，誰にでもパソコン・グラフィックが楽しめるように簡易言語化している．つまり，BASICの知識，プログラムの知識を必要としない．

つぎのような特徴がある．

❶コマンド一発で，直線，円，だ円，色塗り，四角，メッシュ，点線が描ける他，グラフィック用の強力コマンドを多数装備．
❷作成画面データをディスクにセーブ，ロード可能．
❸赤，青，緑の3原色を1,000通りの組み合わせで混ぜ合わせ可能．
❹ハードコピー可能．
❺トレース・モード，BASICジュネレート機能あり．
❻漢字，ひらがな，数字，記号の色指定，サイズ指定可能．

これにより以下の利用が考えられる．

1)グラフィック・デザイナーの配合シミュレーションとして．
2)デザインコンペの記録保存プレゼント用に．
3)簡易ポスターデザインのツールとして．
4)ビジュアルな図形資料として，会議用，レポート提出に．
5)アプリケーション・プログラムメッセージ画面設定ツールとして．
6)教育用グラフィック・デザイン・プログラムとして．

GTシステムの概用

GTは大別してGRAPHIC TOOLとGRAPHIC DRAWERに分かれる．システムが起動すると，2つのモードがPF1，PF2キーに割り付けられる．

GRAPHIC TOOLは数々のコマンドを持ち，メイン・プログラムとも呼べる機能を持つ．一方，GRAPHIC DRAWERは主に，10キーによるドローイングのランダムな曲線を描くときに使うプログラムになっている．

また，スクリーン・セーバー，ローダーは主に，GRAPHIC DRAWERで作られた絵をセーブ，ロードするものに使う．そして，GRAPHIC TOOLを使って作成した絵の上に，GRAPHIC DRAWERで描き加えることは可能だが，この逆はできない．ほとんどのグラフィックはGRAPHIC TOOLで作成できる．

GRAPHIC TOOL

まず，システムにディスケットを入れ，GTを起動する．するとメニュー画面が表われる．ここでPF1キーを押すと，GRAPHIC TOOLが起動する．起動したら，640×200か640×400ドットかを選択する．この後，背景色を指定すれば，種々の絵が描ける．

表1に示すコマンドがある．このうち，PAINTとTILEの違いは，PAINTはRGBの一般的な7色であるのに対し，TILEはRGBの混ぜ合わせにより，1,000の中間色を作れる．

この他，いろいろ便利なコマンドがあるので，紹介する．

H(HOLIZONTAL LINE)：移動カーソル位置を通る水平線を引く．

V(VERTICAL LINE)：移動カーソル位置を通る垂直線を引く．

E(ELLIPSE)：起点カーソルと移動カーソルとの距離を半径とするだ円を描く．歪率は1～1000までの数値で入力．

X(X DIVIDE)：起点カーソル位置と移動カーソル位置を対角線として，横の分割をする．ドット幅を数値で入力する．

Y(Y DIVIDE)：Xと同様に縦を分割する．

W(LINE STYLE)：起点カーソルと移動カーソル間に10種の中の1種類の線を引く．

M(MESH)：起点カーソルと移動カーソルを対象線として，メッシュを描く．このコマンドは方眼紙の役割りをするもので，トレース，ジェネレートは不可．

GRAPHIC DRAWER

これは，GRAPHIC TOOLで描いた絵の細部を変更，修正するときに使う．これで変更，修正した場合，トレース，ジェネレートはできない．作った絵はセーバーでセーブしておくが，このときかなりのディスク容量が必要となるので，注意を要する．

サポート体制

発売以後わかったバグ，ハードのバージョン・アップなどで不都合が生じた場合，新しい応用例などを，GTユーザーズ・ニュースとしてユーザーに配付する．

図1　グラフィック作成画面

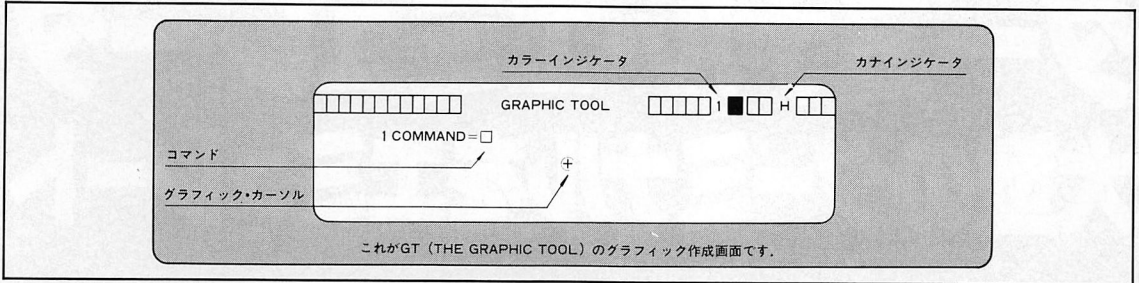



図2　グラフィック・ドロアー作成画面

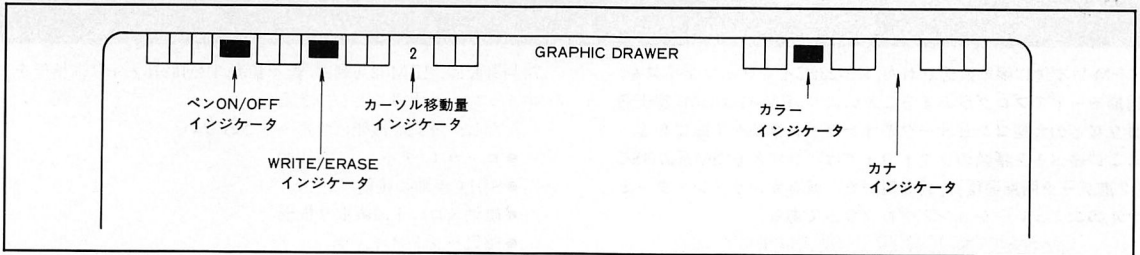



図4　メニュー画面

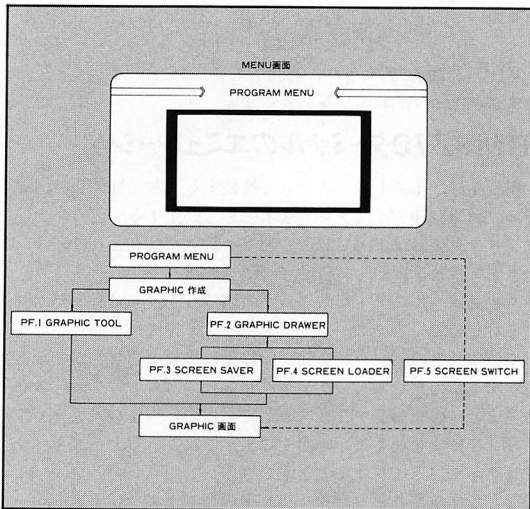
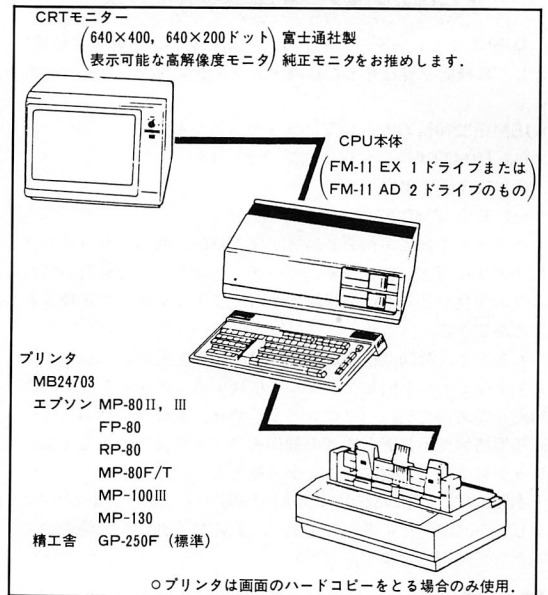



表1　GTコマンド一覧

キー	コマンド	キー	コマンド
[S]	POINT SET	[F.1]	CHANGE COLOR INDICATOR
[R]	POINT RESET	[F.2]	PROCESS LIST
[C]	CIRCLE	[F.3]	TRACE
[E]	ELLIPSE	[F.4]	FILES
[T]	TILE (B/1,R/2,G/3)	[F.5]	HELP
[P]	PAINT	[F.6]	DELETE
[L]	LINE	[F.7]	GENERATE BASIC SENTENCE
[B]	BOX	[F.8]	TRACE BY STEP
[F]	BOXFULL	[F.9]	LOAD DATA
[K]	KANJI	[F.10]	SAVE DATA
[*]	MOVE POSITION	[^H]	H.COPY
[Z]	PALETTE (F.1)	[ESC]	CANCEL
[W]	LINE STYLE	[^E]	KILL
[CLR]	TEXT CLEAR	[HOME]	GRAPHIC CLEAR
[A]	CHARACTER STRINGS	[M]	MESH
[Y]	Y DIVIDE	[X]	X DIVIDE
[H]	HORIZONTAL LINE	[V]	VERTICAL LINE
[@]	PALETTE INITIALIZE	[Q]	RETURN TO MENU
[G]	GRAPHIC ONLY	[¥]	KATAKANA/HIRAKANA

GTの今後

画面入力の手間を省くデジタイザ，イメージスキャナ，ライトペンなど，また出力の多様化として，カラープリンタ，X-Yプロッタのハンドリング・ルーチンなどを開発中である．

図3　ハードウェア構成図

開発会社名：㈱日本コンピュータ設計
販　売　元：㈱日本コンピュータ設計
問い合せ先：㈱日本コンピュータ設計
〒150 東京都渋谷区松濤1-4-9 サンエルビルB1
☎(03)466-6101
価格：￥10,000
使用言語：BASICおよびマシン語
必要なOS：F-BASIC V4.0 DISK BASIC
供給媒体：5インチ・ディスク
マニュアル：ありp.16
マニュアル別売：なし
サンプル・データ：DEMOデータとしてあり
出荷開始時期：'83年3月
最小システム構成：FM-11DISK付き
CRTモニタ(640×400ドット or 640×200ドット) ハードコピー可能なプリンタ
(プリンタは，ハードコピーを取るときのみ使用)

FM-11 ターミナル・エミュレータ 大型のターミナルをエミュレート

FM-11本体に標準装備された，RS-232Cインターフェイスを，同期モードでプログラムすることにより，FM-11はIBM，富士通，日立などの大型コンピュータとオンライン接続が可能になる．ここに述べる２種類のソフトウェアは，いずれもIBM系のBSC（２進データ同期通信）方式を用いた，既存オンライン・ターミナルのエミュレーション・プログラムである．

I3270エミュレータ

I3270エミュレータは，FM-11にIBM系3270型情報表示装置*としての機能を持たせるためのオンライン・ソフトウェアである．

*IBM系3270情報表示装置には次のものがある．

(a) IBM3270

(b) 富士通 F9526

(c) 日立 T560/27

ホスト・コンピュータと通信回線で結び，問い合せ（インクワイアリ)，照会，情報検索，データ・エントリ，会話型のプログラム開発などの業務を遠隔地から，リアルタイムに処理することができる．

もともと，3270型情報表示装置は，制御装置に８～32台のダム・ターミナルを同軸ケーブルで接続する，クラスタ型がその殆んどであったが，パソコンの高性能化，低価格に伴い，この3270型情報表示装置と同等の動作をソフトウェアのエミュレーションにより，実現することが可能となった．

本製品は，上記の3270をFM-11で実現したもので，ホスト・コンピュータが，BSC手順であれば，FM-11を3270型情報表示装置として使用できます．

特　徴

❶BSC手順，IBM3270ターミナル・エミュレーション
❷9,600BPSの通信スピード
❸定様式/不定様式画面
❹1,920文字(80桁×24行)スクリーン
❺４色/２色カラーの切り替え，またはモノクロ表示
❻システム状況表示
❼ライトペンまたはカーソル・セレクトのサポート
❽プリンタへのローカル・ハードコピー
❾ディスクへの仮想プリンタ出力
❿合成画面の生成
⓫ファイル受信機能

IBM3270との対応

本製品は，IBM3271制御装置(モデル２)，IBM3277表示装置，およびIBM3275制御/表示装置（モデル２）の機能と，IBM3274制御装置，IBM3276制御/表示装置，IBM3278表示装置機能をエミュレートすることができる．

ただし，下記の機能はサポートされない．

- ローカル(チャンネル)接続
- SDLC手順の接続
- 磁気スロット読み取り機構
- 磁気ハンドスキャナ
- Extended Field
- キーボード……ATTN, CURSR　BLINK, ALTCURSR, IDENT, DEV　CHCL, ←←　→→ の各キー
- IBM3270(モデル１)……480字スクリーン
- グラフィックおよび漢字表示機能

IBM3270ターミナルのエミュレーション

本製品は，ホスト・システムと接続するとき，制御装置内蔵型の，IBM3270ディスプレイ端末として動作する．

プリンタ装置は，IBM3270バッファ付プリンタとして動作し，ホスト・システムからのCopy命令を有効にする．

本製品は，ホスト・システムとの間で，RTS-CTSモデム制御を行なっているので，マルチ・ポイント接続が可能．したがって，複数台のパソコンを同一回線上に接続することができる．

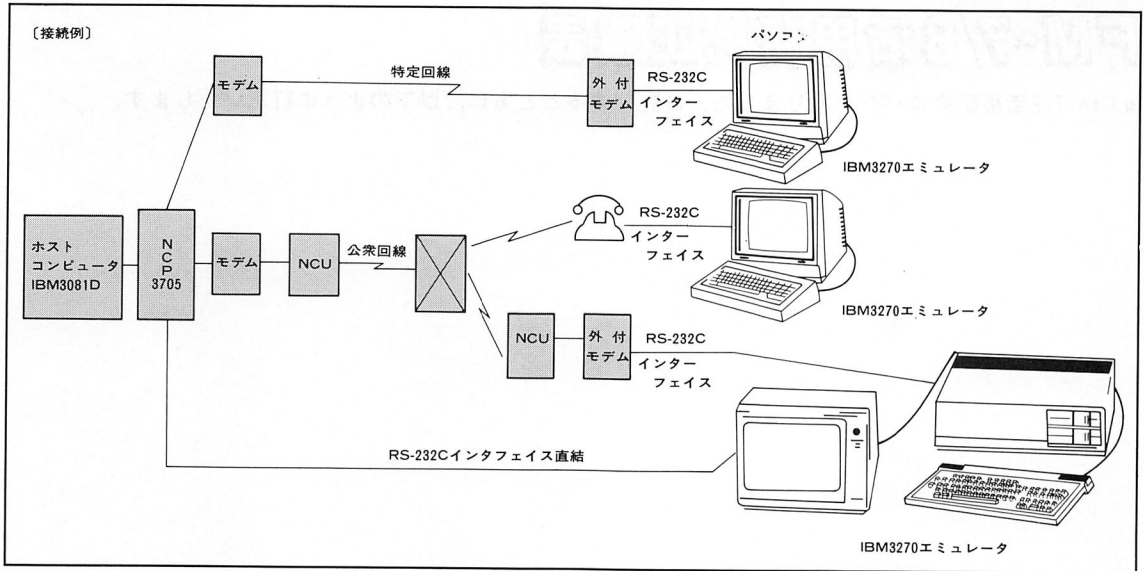
パソコンのフロッピーディスク装置は，ホスト・システムから見ると，仮想第２プリンタとして動作する．また，ローカル・プリント機能を使えば，画面上に表示されているデータを，ダイレクトにファイルすることができる．

ここで作成されるファイルは，そのとき使っているオペレーティング・システム(CP/M-86)のファイル構造になるので，後日，他のアプリケーション・プログラムや，BASIC, COBOLなどで，データ処理が可能になる．

FACOM F9526, HITAC T560/27 ターミナル・エミュレーション

本製品は，富士通Mシリーズのf9525，またはF9526ビデオ・ターミナルのエミュレータとしても動作可能である．F9526は，HDLCとBSC手順がサポートされているが，このエミュレータを使うときはBSC手順となる．

F9526では拡張されたフィールド制御機能（Start Field Extended）があるが，この命令は自動的に無視される．日立Mシリーズと接続する場合は，T560/27(IBM3270モード)で使用可能となる．これらのターミナル・エミュレーションでは，IBM3270エミュレーションとスクリーンなどの動作で一部異なる場合があるが，運用上さほど問題とはならない．

図1

RJEターミナル・エミュレータ

RJEターミナル・エミュレータは，FM-11で実行できる，IBM3780リモートバッチ・ターミナルのエミュレータ・プログラムである．

ホスト・システム（大型コンピュータ）と，パソコンをモデム接続し，パソコンのディスク上に作成したJCL（ジョブ・コントロール・ランゲージ）やPL/I，FORTRAN，COBOLなどのソース・プログラムをホストに送信した後，そのコンパイルや実行結果を受信して，パソコン側のプリンタやディスクへ出力するリモート・ジョブ・エントリに利用できる．

ホストとの伝送制御手段には，IBMに準拠したBSC（2進データ同期通信）プロトコルが採用されており，100%確実なデータの送受信が可能である．

特　長

❶IBM3780，3770，3741ターミナル・エミュレーション
❷BSCI，BSC2プロトコル
❸特定/公衆通信回線
❹IBM3780RJEプロトコルをサポート
- 端末制御文字
- ESCシーケンス
- スペース圧縮/拡張

❺最高9,600ボーの伝送スピード
❻Automatic/Manual受信デバイス・セレクト
❼バイナリ・データの送受信
❽EBCDIC ⟷ JIS(ASCII)コードの自動変換
❾端末 to 端末(パソコン to パソコン)伝送モード
❿Help コマンド
- 『オンライン・モニタ』コマンド
- オートマチック・システム・コンフィギュレーション

システム概要

RJEエミュレータを利用することにより，FM-11は，IBM3780と同機能の通信制御を備えたRJEターミナルになる．

従来の3780では，カードリーダ，ラインプリンタ，そしてカードパンチャが送受信デバイスになっている．RJEエミュレータでは，OS(CP/M-86)がサポートするすべてのI/Oを送受信デバイスの対象にすることができる．

ホストへ送信するJCLやPL/I，FORTRAN，COBOLなどのソース・データは，OS標準のテキスト・エディタでディスク上に作成できる．ただし，JCLに限りコンソールからダイレクトに送信することも可能である．ディスク上に作られたテキスト・データは，自動的にJISからEBCDICコードに変換され，ホストへ送信される．

また，簡単なオプション・スイッチでバイナリ・データとしても送ることが可能である．ホストからの受信データに対しては，バラエティーに富んだ受信機能がサポートとされている．

受信デバイス・セレクタは，ホストから送られてくる『装置選択文字DC1，DC2，DC3』により，オートマチック，あるいはオペレータの任意でコンソール，プリンタ，ディスク，パンチャに選択可能になる．

プリンティング・デバイスへの出力は，ホスト指令による『フォーマット・レコード』とESCシーケンスにより完璧に形式化され，印刷またはディスクに格納される．もちろんバイナリ・データの受信も可能である．この他，RJEエミュレータには，IBM3741モード，あるいはパソコン同士で，データ伝送できるファイル・トランスファ機能や，回線上のトラブルを検出するための『回線モニタ表示』機能が標準装備されている．

開発会社名：㈱インターコム
販　売　元：富士通㈱
問い合わせ：大型コンピュータのSEの方を通して，富士通に問い合わせてください．
価格：I3270エミュレータ　￥170,000　(CP/M-86)
　RJEターミナル・エミュレータ
　　FBASIC用　￥150,000
　　CP/M-86用　￥170,000
使用言語：アセンブラ
必要なOS：CP/M-86またはFBASIC V.4
提供媒体：5インチ・ディスク
マニュアル：あり，約p.80
マニュアル別売：なし
出荷開始時期：CP/M-86版　'84年2月予定
最小システム構成：FM-11EX
　高解像度カラーCRT
　ビジネス・プリンタ
　漢字ROMカード

FM-7/8活用研究正誤表

●FM-7/8活用研究にバグがありました．お詫びするとともに，以下のように訂正いたします．

音声モニタ (p.12)

DISK版の文字が，タイトルから抜けていました．

SUPER MONITOR (p.17)

以下のアドレスのデータを，修正してください．$33D6～$33D8の3バイトを12，$4273～$4284までを8C，0C，5D，10，27，00，99，6D，A0，2A，FC，5A，20，F4，2C，58，2B，20，$4305を2C，$4799をE7に変更します．なお，すべて16進数で$マークが付いているのが，アドレスです．

MEMORY CHENGRの FM-7の変更箇所 (p.26)

リスト1を次のように変更してください．

リスト1

```
310C-BD   FD9E   JSR    $FD9E
310F-86   37     LDA    #$37
3111-BD   EDCB   JSR    $EDCB
3114-BD   FD6E   JSR    $FD6E
3117-FD   D383   STD    $D383
311A-81   12     CMPA   #$12
311C-26   03     BNE    $3121
311E-BD   EAE0   JSR    $EAE0
3121-39          RTS
```

エディタ・アセンブラV2.0 (p.27)

アドレスの$5B87を4Eにしてください．

サウンド・エディタVer.2.0 (p.36)

追加部分：14850 LOADM "EAT" を14850 LOADM "EDT" に．また，p.37の3行目ED7をEDTにし，7行目70FFを7DFF，50FFを5DFFにしてください．p.40リスト1中11000行はREM文にするか，なくしてください．

BASIC ANALYZER (p.66)

表2中BIOSは間接ジャンプでアドレスはいずれもFBFAです．また，アセンブル・リスト中42行目は，8Cの後に07，01を加え，210～222行，989～993行はソース・リストのASCIIコードを書きます．1139～1143行は右のコードを同様に入れます．

DISK SAVER & LOADER (p.87)

リスト3はSAVERアセンブル・リスト，リスト4がLOADERアセンブル・リストです．

Kコンパイラ用グラフィック・サブルーチン (p.134)

リスト中1110，1120行の$BD，$F2D8を$ADFF#，$FBFA#にしてください．

スーパーパターン・エディタ (p.137)

マシン語部アセンブル・リスト中114行の後にSTA ，X+（マシン語はA780）を追加します．

追加版Kコンパイラ用 グラフィック・サブルーチン (p.136)

1140行のABS(%2-7))のカッコが1つ多いので，1つ取り除き，1160行…POKE FD38+%1，%2…に変更します．

Kコンパイラによる日本語ワード・プロセッサ (p.145)

6行～7行	できない場合は	→でない場合は
p.146 各キーの説明		
(3) ファンクション・キー		→ファンクション・キー
(11)【CTRL】−【B】		→【CTRL】−【G】
(14)【CTRL】−【Q】		→【CTRL】−【O】
ローマ字の入力		
(2) "N"(ん)		→"N　"(ん)
"A"="あ"		→"　A"="あ"
"I"="い"		→"　I"="い"
印字する文字の大きさ方向の指定		
4…倍角度指定		→4…倍角指定
特殊記号		
(3) 改行		→改行しない
p.147 図4		
$50A7	起動後は破壊と出る	→起動後は破壊される
BADR+4'	文字数(1)	→最大行数(1)

F-BASIC Ver.1.0をFM-7で (p.169)

左最下段に『なお，このプログラムはFM-8活用研究の"FM-8有象無象"』を追加してください．

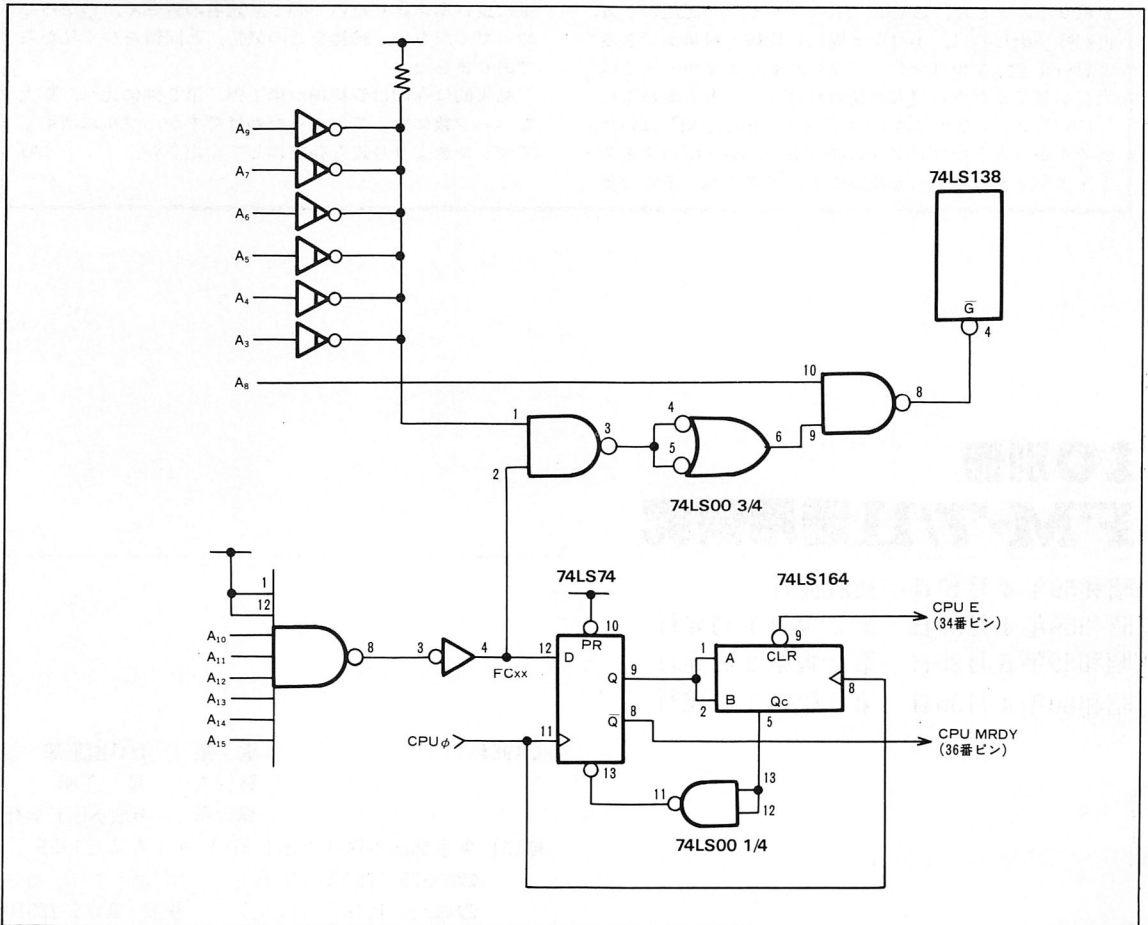
FM-7/8活用研究正誤表

2 dot SCROOL (p.235)

FM-7のとき$6F83と$6F96を1Fにしてください.

HERO-09の改造 (p.213)

TTLピン接続図 (p.240)

1部ピン番号がおかしなものがありますが，すべて左下が1番で反時計まわりに番号が付きます.

編集後記

■FM活研第3弾『FM-7/11活用研究』いかがでしたか．この本に何を入れるかで，七転八倒，苦しみ悶え，あれも入れたい，これも入れたい，でもページ数が…，ウォー！と叫び，編集部内を駆けずり廻ってやっとできたのがこの本です．

FM-11のユーザーの方には必読の資料も，もっと入れたかったのです．サブシステム・コマンド一覧もページの都合でタイマ関係などが抜けてしまっています．ごめんなさい．BIOSやディスプレイ・サブシステムの解説書は，富士通のアブソリュート・アセンブラについています．

FM-7用のIODOS 9は，作者sgnさんによると，『Kコンパイラのユーザーに送るIODOS 9』で，

- Kコンパイラで作成したプログラムが，すぐにコマンドになる．フリーエリアは43Kバイトで，大きなプログラムでもコンパイル可能．
- Kコンパイラのユーザーでなくても，52Kバイトのフリーエリアを自由に使える．
- ロード，セーブなど，F-BASIC感覚で使える．

などです．ディスケット・サービス時には，この本ののより機能アップしていると思います．

TortoiseをBASICとKコンパイラ両方で記述したものを載せてみました．BASICとKコンパイラの記述のし方，再起呼び出しの可，不可など面白い比較・対象ができると思います．多少コマンドに違いがありますが，そこは大目に見てください（人が変わればコマンドも変わる）．

UNIXライクなテキスト・エディタ〝micro vi〟はいかがですか．大きなプログラムになると，BASICのテキスト・エディタはちょっと使いづらいですよね．そんなとき，この〝micro vi〟でエディットしてみてください．

FM-11の〝グラフィック・エディタ〟，すごいですね．話題のC言語です．これだけの量のCのソースを発表するのはたぶん，この本が初めてだと思います．C言語を勉強中の方，どんな言語だろうと思っている方は，大いに参考にしてください．

ゲーム作りの好きな方に，〝MUSIC MAKER〟は強い味方になってくれます．ゲームには，音楽や効果音は必要不可欠ですからね．このソフトの面白いところは，BASICがPSGに送ったコードをノゾキミて，コードをメモリに書き込むことですね．横取りができるのは，IRQ割り込みで行なっているためですね．逆アセンブルしてみたらどうでしょう．

このような本が作れるのも，ユーザーの方々が，日頃どうマイコンを使いこなそうか，どんなソフトを作ろうか，など色々模索しているからだと思います．そして，その結果が本書です．ユーザーの方々の力量には感嘆します．これからも，どんどんプログラムを作ってI/Oに投稿してください．　(K_1)

■活研ファンの皆さん，お待たせしました．本当はもっと早く出版できると思っていたのですが，いざ取り組んでみると原稿の選択（これが本当に時間がかかる），リストの変更など，内容が揃うまでにすでに日数が…．その後もまだ色々手間のかかることばかり．でも，これを一重に良い本を作りたい一心で．読者の皆さん，わかってやってください，純粋なこの気持．と同情をひく私たちでありました．

結果的に今回は資料編が第1弾，第2弾に比べ，膨大なページ数になってしまったわけですが，フルに活用してマシンをより身近な存在にしてください．　(A)

I/O別冊
FM-7/11活用研究

昭和59年4月10日　初版発行
昭和59年4月30日　第2版第1刷発行
昭和59年5月25日　第2版第2刷発行
昭和60年4月30日　第3版第1刷発行

編　集　I/O編集部
発行人　星　正明
発行所　株式会社工学社
〒151 東京都渋谷区代々木1-37-1 ぜんらくビル5F
☎(03)375-5784(代)〔営業〕
☎(03)320-1218(代)〔編集〕　振替 東京5-22510

印刷：㈱耕文社

定価2500円

雑誌61473-60